

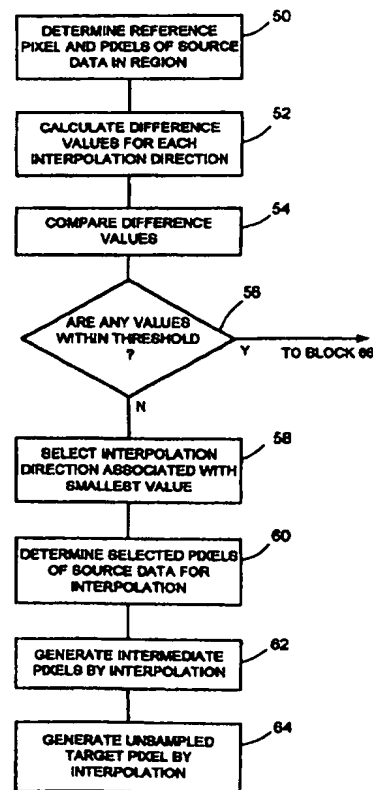


INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : H04N 5/262	A1	(11) International Publication Number: WO 97/17801 (43) International Publication Date: 15 May 1997 (15.05.97)
(21) International Application Number: PCT/CA96/00703 (22) International Filing Date: 22 October 1996 (22.10.96) (30) Priority Data: 08/555,289 8 November 1995 (08.11.95) US (71) Applicant (for all designated States except US): GENESIS MICROCHIP INC. [CA/CA]; Suite 400, 200 Town Centre Boulevard, Markham, Ontario L3R 8G5 (CA). (72) Inventors; and (75) Inventors/Applicants (for US only): GREGGAIN, Lance [CA/CA]; 32 Woburn Drive, Woodbridge, Ontario L4L 7H8 (CA). NGO, Calvin [CA/CA]; 175 Royal Orchard Boulevard, Thornhill, Ontario L3T 3E1 (CA). (74) Agent: PERRY, Stephen, J.; Sim & McBurney, 6th floor, 330 University Avenue, Toronto, Ontario M5G 1R7 (CA).		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, US, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG). Published <i>With international search report.</i>

(54) Title: METHOD AND APPARATUS FOR VIDEO SOURCE DATA INTERPOLATION**(57) Abstract**

A method of generating an upsampled target pixel positioned between two lines of input source data includes the step of comparing pixels of different lines of the source data in a region surrounding the upsampled target pixel to be generated in at least two different directions (50, 52, 54). An interpolation direction based on the comparison is selected (58) and interpolations between selected pixels of the source data in the determined interpolation direction are carried out to compute intermediate pixels on a line segment passing through the upsampled target pixel (60, 62). An interpolation between the intermediate pixels is carried out to generate the upsampled target pixel (64). An apparatus for performing the method is also disclosed.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgyzstan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Latvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			VN	Viet Nam

METHOD AND APPARATUS FOR VIDEO SOURCE DATA INTERPOLATION

TECHNICAL FIELD

The present invention relates to data interpolation and in particular to a method of generating an upsampled target pixel from input source data and an apparatus for performing the method.

BACKGROUND ART

Video images can be represented as a two-dimensional array of digital data with each data point in the array representing a pixel of the digitized video image. The value assigned to each pixel determines its intensity and/or colour when the video image is recreated. In the video imaging art, it is often desired to magnify a video image to arbitrary magnification factors to create enlarged video images. When creating an enlarged image from an original image represented by a digital data array, it is necessary to interpolate between pixels of the original digital data array to generate upsampled pixels between consecutive lines and pixels of the original digital data array to "fill" in pixels of the enlarged image. Prior art techniques to create these enlarged video images have been considered.

It is common practice in the prior art to use an orthogonal, Cartesian approach to interpolate separately the vertical and horizontal dimensions of the digitized video image to be enlarged. However, this technique typically results in enlarged video images which suffer from jagged edges commonly referred to as "stairstepping". Improvements in the interpolation method used may reduce the stairstepping, but it cannot be completely avoided because it is inherent when using this technique.

A directional interpolation scheme addresses the stairstepping problem. Directional interpolation recognizes the geometric structure in images. With directional interpolation, it has been found that interpolation along the edges produces improved results over interpolation across the edges. Directional interpolation involves analysing local image structure and performing the interpolation based on the image structure.

U.S. Patent No. 5,019,903 to Dougall et al. discloses an apparatus for directionally interpolating between lines of a supersampled digital signal. The apparatus calculates gradient vectors for use in directionally interpolating a vertically upsampled line. This operation makes the apparatus best suited for line-doubling deinterlacing applications.

U.S. Patent No. 5,347,599 to Yamashita et al. discloses an adaptive interpolation method and an apparatus using correlation detection. The apparatus performs calculations on lines of pixels of the source data to select the desired upsampling direction. However, similar to Dougall et al, this apparatus is also best suited for line-doubling deinterlacing applications.

Although Dougall et al. and Yamashita et al. disclose alternative directional interpolation methods to generate upsampled pixels, their design is such as to support only fixed, integral vertical resize factors. Accordingly, there exists a need for an improved interpolation method and apparatus that supports arbitrary, fractional resize factors in both vertical and horizontal dimensions.

It is therefore an object of the present invention to provide a novel method and apparatus for generating an upsampled target pixel from input source data.

DISCLOSURE OF THE INVENTION

The present invention performs directional interpolation to arbitrary, fractional resize factors. First, an interpolation direction is selected corresponding to any low-frequency edges in the source data. Next, directional interpolation is performed to generate intermediate pixels. The intermediate pixels are then interpolated, possibly non-orthogonally, to generate the desired upsampled pixel.

More specifically, according to one aspect of the present invention there is provided a method of generating an upsampled target pixel from input source data comprising the steps of:

- (i) comparing pixels of different lines of said source data in a region surrounding an upsampled target pixel to be generated in at least two different directions;

- 3 -

- (ii) selecting an interpolation direction based on the comparisons of step (i);
- (iii) interpolating between selected pixels of different lines of said source data in the interpolation direction determined at step (ii) and computing
5 intermediate pixels on a line segment passing through said upsampled target pixel; and
- (iv) interpolating between the intermediate pixels to generate said upsampled target pixel.

10 In the preferred embodiment, the line segment is parallel to the lines of the source data. Restricting the line segment to be horizontal as such, greatly simplifies the calculation of the intermediate pixels and the upsampled target pixel.

15 In one possible embodiment, pixels of different lines of source data are compared in three different directions, including a vertical direction, and opposite oblique directions forming an angle with the vertical direction that is in the range $0 < \text{angle} < 90^\circ$. During the comparisons of step (i), difference values between the compared pixels are generated, the difference values being used to select the interpolation direction at step (ii).

20 In this embodiment, prior to step (ii), the difference values resulting from the comparisons in the three directions are compared to determine whether the differences therebetween are within a threshold. The direction associated with the smallest difference value is selected as the interpolation direction when none of the differences are within the threshold. When the differences between the difference values resulting from the comparisons in all three directions are within the threshold, the vertical direction is selected as the interpolation direction. When the
25 difference between the difference values resulting from the comparisons in the vertical direction and only one of the oblique directions is within the threshold, that oblique direction is selected as the interpolation direction. When the difference between the difference values resulting from the comparisons in the two oblique directions is within the threshold, additional pixels of the different lines of source
30 data are compared in the opposite oblique directions. The difference values generated as a result of these comparisons are compared. When the difference between the difference values resulting from the comparisons of the additional

- 4 -

pixels is within the threshold, the vertical interpolation direction is selected as the interpolation directions.

5 In another embodiment the line segment is once again parallel to the lines of the source data but this time pixels of the different lines of source data are compared in seven directions at step (i). These directions include the vertical direction and three pairs of opposite oblique directions forming varying angles with the vertical direction that are within the range $0^\circ < \text{angle} < 90^\circ$. Difference values between the compared pixels of step (i) are calculated.

10 In this embodiment, the set of difference values to the left of the set under consideration and the set to the right are also examined. The three difference values for each possible oblique interpolation direction are summed and compared to determine if the differences therebetween are within a threshold. When none of the differences are sufficiently small to indicate a low-frequency edge in the source image, then the vertical direction is selected as the interpolation direction. When
15 none of the differences are within the threshold, then the oblique direction associated with the smallest difference value is selected as the interpolation direction. When difference values associated with a "right-ward" oblique direction and a "left-ward" oblique direction are within the threshold, then the vertical direction is once again selected as the interpolation direction. Otherwise one of the
20 oblique directions is selected as the interpolation direction.

According to another aspect of the present invention there is provided an apparatus for generating an upsampled target pixel from input source data comprising:

25 comparison means for comparing pixels of different lines of said source data in a region surrounding an upsampled target pixel to be generated, in at least two different directions;

selection means for selecting an interpolation direction in response to said comparison means; and

30 upsampling means for interpolating between selected pixels of different lines of said source data in the interpolation direction selected by said selection means to compute intermediate pixels on a line segment passing through

- 5 -

said upsampled target pixel and for interpolating between the intermediate pixels to generate said upsampled target pixel.

The present invention provides advantages in that upsampled target pixels can be generated to allow large magnified images to be created with reduced
5 stairstepping, particularly in the case of large magnification factors.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the present invention will now be described more fully with reference to the accompanying drawings in which:

Figure 1 is a diagram showing the pixels of three consecutive lines
10 of video image source data, the position of an upsampled target pixel to be generated and the region surrounding pixels of the source data used to generate the upsampled target pixel;

Figure 2 is a diagram showing the comparisons that are made
15 between pixels of the source data in order to determine the interpolation direction to be used when generating an upsampled target pixel;

Figure 3 is a diagram illustrating additional comparisons that are made between pixels of the source data when comparisons in two oblique directions yield similar values;

Figure 4 is a diagram showing a vertical upsampling technique used
20 to generate an upsampled target pixel;

Figure 5 is a diagram showing a negative slope oblique upsampling technique used to generate an upsampled target pixel;

Figure 6 is a diagram showing a positive slope oblique upsampling technique used to generate an upsampled target pixel;

25 Figures 7a and 7b are flowcharts showing the steps performed in the first method when generating an upsampled target pixel from the source data;

Figure 8 is a block diagram showing an apparatus for generating upsampled target pixels from input source data;

30 Figure 9 is a schematic of a delay circuit forming part of the apparatus illustrated in Figure 8;

- 6 -

Figure 10 is a schematic of a decision circuit forming part of the apparatus illustrated in Figure 8;

Figure 11 is a schematic of a β calculator forming part of the apparatus illustrated in Figure 8;

5 Figure 12 is a schematic of an oblique upsampler forming part of the apparatus illustrated in Figure 8;

Figure 13 is a diagram illustrating the comparisons between source pixels made in determining the interpolation direction for a second embodiment;

Figure 14 is a diagram illustrating a boundary crossing situation;

10 Figure 15 is a diagram showing a vertical upsampling technique used to generate an upsampled target pixel;

Figure 16 is a diagram showing a positive 45° slope oblique upsampling technique used to generate an upsampled target pixel;

15 Figure 17 is a diagram showing a negative 45° slope oblique upsampling technique used to generate an upsampled target pixel;

Figure 18 is a diagram showing the source pixels used in a 26.6° slope oblique upsampling technique.

Figure 19 is a diagram showing a positive 26.6° slope oblique upsampling technique used to generate an upsampled target pixel;

20 Figure 20 is a diagram showing a negative 26.6° slope oblique upsampling technique used to generate an upsampled target pixel;

Figure 21 is a diagram showing a positive 63.4° slope oblique upsampling technique used to generate an upsampled target pixel;

25 Figure 22 is a diagram showing a negative 63.4° slope oblique upsampling technique used to generate an upsampled target pixel;

Figures 23a and 23b are flowcharts showing the steps performed in a second method when generating an upsampled target pixel from the source data;

Figure 24 is a block diagram showing a second apparatus for generating upsampled target pixels from input source data;

30 Figure 25 is a schematic of a delay circuit forming part of the apparatus illustrated in Figure 24;

- 7 -

Figure 26 is a schematic of a decision circuit forming part of the apparatus illustrated in Figure 24;

Figure 27 is a schematic of a β calculator forming part of the apparatus illustrated in Figure 24; and

5 Figure 28 is a schematic of an oblique upsampler forming part of the apparatus illustrated in Figure 24.

BEST MODE FOR CARRYING OUT THE INVENTION

When upsampling input video image digital source data to create an enlarged video image, it is necessary to generate upsampled target pixels between
10 the lines and pixels of the digital source data. The resize factor and the resize direction (i.e. vertical and/or horizontal) determine the number as well as the position of the upsampled target pixels to be generated.

In order to reduce stairstepping and create enlarged video images of high resolution, it is necessary to calculate accurately the values for the various
15 upsampled target pixels to be generated. Thus, the upsampling interpolation direction used to generate the values of the upsampled target pixels should be selected to achieve this goal. Failure to select an upsampling interpolation direction which results in accurate values for the upsampled target pixels will result in the generation of enlarged images with noticeable stairstepping.

20 The present method and apparatus generates an upsampled target pixel by calculating the difference in values of pixels of the digital source data in a region surrounding the upsampled target pixel to be generated in a number of directions. The calculated difference values are then examined in order to select an interpolation direction. Interpolations between selected pixels of the digital source
25 data in the determined interpolation direction are then performed to generate intermediate pixels on a line segment that passes through the upsampled target pixel to be generated. Restricting the line segment to be parallel to the source data – that is, horizontal – simplifies the calculation of the intermediate pixels. Interpolations between the intermediate pixels are then performed to generate the
30 upsampled target pixel.

An embodiment of the present method of generating an upsampled target pixel from input source data will now be described with particular reference to Figures 1 to 7b.

When an upsampled target pixel w is to be generated, the closest pixel to the upsampled target pixel in the digital source data is determined and is used as a reference pixel P_R in the interpolation method. The two consecutive lines of source data between which the upsampled target pixel is located are also determined. The reference pixel, the adjacent pixel on the same line of source data as the reference pixel P_R and the two pixels on the other line of source data that form a rectangular region R surrounding the upsampled target pixel are determined (see block 50 in Figure 7a). The four source pixels that define the boundary of region R are used initially to determine the interpolation direction as will be described. The region R may correspond to any one of the four quadrants, namely top-left, top-right, bottom-left or bottom-right surrounding the reference pixel P_R .

Figure 1 shows a small segment of digital source data. The source pixels are labelled relative to the reference pixel $P_R = U_{00}$. The first line of source data is represented by five pixels $U_{-1,2}$ to $U_{1,2}$, the second line of source is represented by pixels $U_{0,2}$ to $U_{0,2}$ and the third line of source is represented by pixels $U_{1,2}$ to $U_{1,2}$. Clearly, pixel U_{00} is the closest pixel to upsampled target pixel w and is the designated reference pixel P_R . In this example, the upsampled target pixel w is positioned between the first and second lines in the top-left quadrant of region R . Pixels $U_{-1,1}$, $U_{-1,0}$, $U_{0,1}$ and $U_{0,0}$ are the pixels that define the region R surrounding the upsampled target pixel w and are used in the interpolation direction making decision. The following chart shows the pixels of the source data that define the region R in the four quadrants surrounding the reference pixel:

<u>Quadrant</u>	<u>Pixels in Region R</u>
Top-Left	$U_{00}, U_{0,1}, U_{-1,1}, U_{-1,0}$
Top-Right	$U_{00}, U_{0,1}, U_{1,1}, U_{1,0}$
Bottom-Left	$U_{00}, U_{0,1}, U_{-1,1}, U_{-1,0}$
Bottom-Right	$U_{00}, U_{0,1}, U_{1,1}, U_{1,0}$

Once the four pixels of region R have been determined, calculations are performed corresponding to three interpolation directions (see block 52). In the example of Figure 1, region R is defined by pixels: $U_{-1,-1}$, $U_{-1,0}$, $U_{0,-1}$ and $U_{0,0}$ and the following calculations are performed:

5	<u>Calculation</u>	<u>Associated Interpolation Direction</u>	
	$\text{abs}[U_{0,0}-U_{-1,-1}]$	oblique 45° line up-left	O_L (1)
	$\text{abs}[U_{0,-1}-U_{-1,0}]$	oblique 45° line up-right	O_R (2)
	$\text{abs}[U_{0,0}-U_{-1,0}]$	vertical	V (3)

Figure 2 illustrates the interpolation directions O_L , O_R and V associated with the three calculations performed on pixels $U_{-1,-1}$, $U_{-1,0}$, $U_{0,-1}$ and $U_{0,0}$. Similar calculations apply for the other three quadrants.

The calculation which yields the smallest difference value is usually associated with the interpolation direction which will yield an upsampled target pixel w having a value which is most similar to the values of the surrounding pixels of the source data. However before the interpolation direction is selected based on the results of these calculations, the calculated difference values are compared with each other to determine whether the differences are within a threshold of one another (see blocks 54 and 56).

If one of the calculated difference values is clearly smaller than the other two values, the interpolation direction associated with the smallest calculated value is selected (block 58). If all three calculated difference values are within the threshold, the vertical interpolation direction V is selected (blocks 66 and 68).

If the calculated difference value associated with the vertical interpolation direction V and the calculated difference value associated with only one of the oblique directions O_L or O_R are within the threshold of one another, the oblique interpolation direction is selected (blocks 70 and 72).

In the case where the calculated difference values associated with the two oblique interpolation directions O_L and O_R are within the threshold of one another, it is important to select the proper interpolation direction since an incorrect decision will yield an enlarged image with noticeable discontinuities in diagonal lines. To assist in the selection process, calculations using additional pixels from

- 10 -

the same lines of source data are performed (block 74). Figure 3 shows these additional calculations for the example of Figures 1 and 2:

	<u>Calculation</u>	<u>Associated Interpolation Direction</u>		
	$\text{abs}[U_{00}-U_{11}]$	oblique 45° up-right	$O1_{AR}$	(4)
5	$\text{abs}[U_{01}-U_{12}]$	oblique 45° up-left	$O1_{AL}$	(5)

These additional calculated difference values are then compared to determine if they are within the threshold of one another (block 76). If one of these difference values is clearly smaller than the other, the oblique interpolation direction O_L or O_R associated with that calculated difference value is selected (block 58). However, if these two additional calculated difference values are within the threshold of one another, the additional following calculations using additional pixels from the same lines of source data are performed (block 78):

	<u>Calculation</u>	<u>Associated Interpolation Direction</u>		
	$\text{abs}[U_{02}-U_{11}]$	oblique 45° up-right	$O2_{AR}$	(6)
15	$\text{abs}[U_{01}-U_{10}]$	oblique 45° up-left	$O2_{AL}$	(7)

Similarly, these additional calculated difference values are compared to determine if they are within the threshold of one another (block 80). If one of these difference values is clearly smaller than the other, the oblique interpolation direction O_L or O_R associated with that calculated difference value is selected (block 58). However, if these two additional calculated difference values are within the threshold of one another, the vertical interpolation direction V is selected as a default to avoid the resolution problems which would occur if the incorrect oblique interpolation direction was selected (block 82).

Once the interpolation direction is determined, selected pixels of the source data are determined based on the interpolation direction (block 60) and the interpolation direction is used to interpolate between the selected pixels of the source data to generate intermediate pixels v (block 62). The intermediate pixels are then interpolated horizontally to generate the upsampled target pixel w (block 64).

Before the upsampled target pixel w can be generated, the location of the upsampled target pixel w must be determined. The upsampled target pixel w is defined by its distances from the reference pixel P_R both in the vertical and horizontal directions. These distances are represented by the terms α_y and α_x where:

$$-0.5 < \alpha_y \leq 0.5; \text{ and}$$

$$-0.5 < \alpha_x \leq 0.5;$$

α_y being positive downward and α_x being positive to the right. The signs of α_y and α_x are related to the quadrant as follows:

Quadrant	α_x	α_y
Top-Left	$\alpha_x < 0$	$\alpha_y < 0$
Top-Right	$\alpha_x \geq 0$	$\alpha_y < 0$
Bottom-Left	$\alpha_x < 0$	$\alpha_y \geq 0$
Bottom-Right	$\alpha_x \geq 0$	$\alpha_y \geq 0$

With the values of α_x and α_y known, interpolations are performed between the selected pixels of the source data in the determined interpolation direction. The resulting intermediate pixels v_i are positioned on a horizontal line segment which passes through the upsampled target pixel w . In the present embodiment linear upsampling is described, however higher orders of interpolation can be used.

Figure 4 shows an upsampled target pixel w positioned in the bottom-right quadrant. In this case, both α_x and α_y are positive. The region R is defined by pixels U_{00} , U_{01} , U_{10} and U_{11} and pixel U_{00} defines the reference pixel P_R . Intermediate pixels v_0 , v_1 and v_2 are also shown and have been computed using linear upsampling in the vertical direction. The intermediate pixels v_0 , v_1 and v_2 in this case are computed using the following equations:

$$v_0 = (1 - \alpha_y)U_{0,1} + \alpha_y U_{1,1} \quad (8)$$

$$v_1 = (1 - \alpha_y)U_{00} + \alpha_y U_{10} \quad (9)$$

$$v_2 = (1 - \alpha_y)U_{01} + \alpha_y U_{11} \quad (10)$$

- 12 -

After the three upsampled intermediate pixels have been computed, interpolations are performed orthogonally horizontally between the two intermediate pixels on opposite sides of the upsampled target pixel w to determine the value of the upsampled target pixel. Therefore, before the upsampled target pixel w can be computed, it is necessary to determine which two intermediate pixels to select. In order to select the intermediate pixels, the term β is defined as the horizontal distance from the middle intermediate pixel v_1 to the upsampled target pixel w . The term β is positive to the right. For vertical interpolation, $\beta = \alpha_x$. The sign of β determines which two intermediate pixels surround the target pixel w . In the example of Figure 4, $\beta \geq 0$ so the interpolation is performed between v_1 and v_2 . The value of the upsampled target pixel w is computed using the following equation:

$$w = (1 - \alpha_x)v_1 + \alpha_x v_2 \quad (11)$$

In other quadrants, that is, cases for α_y and α_x , the intermediate pixels v and the upsampled target pixel w are calculated in a similar manner using different pixels of the source data that surround the reference pixel P_R .

When the oblique up-left interpolation direction O_L is selected, the intermediate pixels v_0 , v_1 and v_2 are computed by interpolating linearly between selected pixels of the source data along negatively sloped diagonal lines. When linearly interpolating, the important parameter is the ratio of the distance of the interpolated pixel endpoints. Since the lines of source data and the horizontal line segment defined by the intermediate pixels are parallel, the intermediate pixels can be defined by α_y , similar to the vertical upsampling case.

Figure 5 shows the case of an upsampled target pixel w positioned in the bottom-right quadrant with α_y and α_x once again positive. Region R is defined by pixels U_{00} , U_{01} , U_{10} and U_{11} with pixel U_{00} defining the reference pixel P_R . Intermediate pixels v_0 , v_1 and v_2 are also shown and have been computed using oblique up-left interpolation. The intermediate pixels v_0 , v_1 and v_2 are computed using the following equations:

- 13 -

$$v_0 = (1-\alpha_y)U_{0,1} + \alpha_y U_{1,0} \quad (12)$$

$$v_1 = (1-\alpha_y)U_{0,0} + \alpha_y U_{1,1} \quad (13)$$

$$v_2 = (1-\alpha_y)U_{0,1} + \alpha_y U_{1,2} \quad (14)$$

After the three upsampled intermediate pixels have been computed, interpolations are performed non-orthogonally horizontally between the two intermediate pixels on opposite sides of the upsampled target pixel w to determine the value of the upsampled target pixel. In the example of Figure 5, since upsampling has been performed along 45° diagonal lines, the horizontal distance from intermediate pixel v_1 to reference pixel $U_{0,0}$ is the same as the vertical distance, namely α_y . Thus, the distance β from the upsampled target pixel w to the intermediate pixel v_1 is equal to $\alpha_x - \alpha_y$. In fact this equation for β in oblique up-left upsampling is valid for all quadrants. β is negative in this case implying that the interpolation is between intermediate pixels v_1 and v_0 . Therefore, the value of the upsampled target pixel w is computed using the following equation:

$$w = (1-(-\beta))v_1 + (-\beta)v_0 \quad (15)$$

In other quadrants, the intermediate pixels v and the upsampled target pixel w are calculated in a similar manner using different selected pixels of the source data that surround the reference pixel P_R .

When the oblique up-right interpolation direction O_R is selected, the intermediate pixels v_0 , v_1 and v_2 are computed by interpolating linearly between selected pixels of the source data located along positively sloped diagonal lines in a similar manner to that described above with reference to Figure 5.

Figure 6 shows an upsampled target pixel w positioned in the region R defined by pixels $U_{0,0}$, $U_{0,1}$, $U_{1,0}$ and $U_{1,1}$ with pixel $U_{0,0}$ defining the reference pixel P_R . Intermediate pixels v_0 , v_1 and v_2 are also shown and have been computed using oblique up-right interpolation. The intermediate pixels v_0 , v_1 and v_2 are computed using the following equations:

$$v_0 = (1-\alpha_y)U_{0,1} + \alpha_y U_{1,2} \quad (16)$$

$$v_1 = (1-\alpha_y)U_{0,0} + \alpha_y U_{1,1} \quad (17)$$

$$v_2 = (1-\alpha_y)U_{0,1} + \alpha_y U_{1,0} \quad (18)$$

- 14 -

After the three upsampled intermediate pixels have been computed, interpolations are performed non-orthogonally horizontally between the two intermediate pixels on opposite sides of the upsampled target pixel w to determine the value of the upsampled target pixel. For oblique up-right upsampling it can be shown that $\beta = \alpha_x + \alpha_y$. In the case of Figure 6, β is positive implying that the interpolation is between intermediate pixels v_1 and v_2 . The value of the upsampled target pixel w is computed using the following equation:

$$w = (1-\beta)v_1 + \beta v_2 \quad (19)$$

In other quadrants, the intermediate pixels v and the upsampled target pixel w are calculated in a similar manner using different selected pixels of the source data that surround the reference pixel P_R .

The above-described method shows access to three lines of source data to demonstrate the cases for the four possible quadrants. It should be noted, however, that once the upsampling quadrant is known only two lines of source data are needed to perform the interpolations and generate the upsampled target pixel.

Referring now to Figure 8, a source data interpolation apparatus for carrying out the above described method is shown and is generally indicated by reference numeral 100. The apparatus 100 receives a stream of input digital source data D_{in} and interpolates between selected pixels U of the source data in order to generate upsampled target pixels w .

The apparatus 100 includes a delay circuit 102 receiving the stream of source data. The delay circuit extracts the required pixels of the source data necessary to generate the desired upsampled target pixel w and passes the required pixels of source data to a decision circuit 104 as well as to an oblique upsampler 106. The decision circuit 104 performs calculations numbered 1 to 3 and 4 to 7 on the pixels of source data if necessary and compares the calculated difference values with the threshold to determine the desired interpolation direction. The interpolation direction output of the decision circuit 104 is fed to the oblique upsampler 106 and to a β calculator 108. The output of the β calculator 108 is applied to the oblique upsampler 106 which uses the input data to generate the desired upsampled target pixel w .

- 15 -

Figure 9 better illustrates the delay circuit 102. As can be seen, the delay circuit has an input line 112 receiving the stream of input source data necessary to generate the desired upsampled target pixel w . The input line branches, with one branch 114 leading directly to three delay elements 120, 122 and 124 connected in series. The other branch 116 leads to a line store 126. The output of the line store 126 leads to three delay elements 128, 130 and 132 connected in series. The line store and the delay elements in each branch allow the delay circuit 102 to output four pixels from two consecutive lines of input source data simultaneously. For the case of interpolating in the bottom-right quadrant, with both α_x and α_y positive, the eight pixels extracted correspond to pixels U_{0-1} , U_{00} , U_{01} , U_{02} , U_{1-1} , U_{10} , U_{11} , and U_{12} as shown.

The decision circuit 104 is best seen in Figure 10 and has six input lines, each of which receives a selected pixel output by the delay circuit 102. The inputs are applied pairwise to subtractors 140, 146, and 150. The outputs of the subtractors are applied through delay elements 142, 144, 148, 152 and 154 to the selection logic 156 and represent the difference value results of calculations numbered 1 to 3. The delay elements are necessary in order provide the difference value results corresponding to calculations numbered 4 to 7. Specifically, the output of subtractor 140, corresponds to $O1_{AR}$, a "future" value of O_R . This value is applied to delay element 142 to give O_R , which is then applied to delay element 144 to give $O2_{AR}$. Likewise, the output of subtractor 150 corresponding to $O2_{AL}$, is applied to delay element 152 to give O_L which is then applied to delay element 154 to give $O1_{AL}$. To "equalize" the latency, delay element 148 is necessary to delay the output of subtractor 146. This value corresponds to V . All these values are passed to the selection logic 156.

The selection logic 156 which receives the α_x and α_y constants as well as the threshold value compares the calculated difference values to determine if any two or all three calculated values are within the threshold. In the cases where clearly one calculated difference value is the smallest, where the calculated difference value associated with the vertical interpolation direction and the calculated difference value associated with only one oblique interpolation direction are both within the threshold or where all three calculated difference values are

- 16 -

within the threshold, the selection logic 156 outputs the selected interpolation direction to the β calculator 108 as well as to the oblique upsampler 106. In the case where the calculated difference values associated with the two oblique directions are within the threshold, the values $O1_{AR}$ and $O1_{AL}$ and possibly $O2_{AR}$ and $O2_{AL}$ are used to determine which interpolation direction to output to the β calculator 108 and to the oblique upsampler 106.

The source pixels shown in Figure 10 correspond to the source pixels used in interpolating in the bottom-right quadrant. These source pixels represent the "future" value of the differences.

The β calculator 108 is best seen in Figure 11 and includes a multiplexer 160 which receives the α_y and $-\alpha_y$ constants as well as the interpolation direction output by the selection logic 156. The multiplexer output is supplied to an adder 162 which also receives the α_x constant. The output of the adder 162 constitutes the β constant output of the β calculator 108.

Figure 12 best illustrates the oblique upsampler 106 and as can be seen it comprises three linear interpolators 106a, 106b and 106c, although in principle higher order interpolations could be used. Each linear interpolator includes two adders and a multiplier. Two linear interpolators are used for directional interpolation and one for horizontal interpolation. The oblique upsampler also includes four multiplexers 165 to 168, which receive the sign of the β constant and four multiplexers 170 to 176, each of which receives the interpolation direction from the selection logic 156. Multiplexers 165 and 166 provide output to multiplexer 170 and multiplexers 167 and 168 provide output to multiplexer 172.

The source pixels input to multiplexers 165 to 168 and multiplexers 174 to 176 illustrate the case for interpolating the bottom right quadrant. Multiplexers 170 and 172 provide output to linear interpolator 106a. Linear interpolator 106a includes an adder 178 receiving input from multiplexers 170 and 172 which supplies input to a multiplier 180. The multiplier 180 which receives the α_y constant provides input to another adder 182 which also receives input from multiplexer 172.

- 17 -

Similarly, multiplexers 174 and 176 provide output to linear interpolator 106b. Linear interpolator 106b includes an adder 184 receiving input from multiplexers 174 and 176 which supplies input to a multiplier 186. The multiplier 186 which receives the α_y constant provides input to another adder 188 which also receives input from multiplexer 176.

The output of the two linear interpolators 106a and 106b respectively are supplied to linear interpolator 106c. Linear interpolator 106c includes an adder 190 receiving input from the adders 182 and 188 which supplies input to multiplier 192. The multiplier 192 receives the β constant from the β calculator 108 and supplies input to adder 194. Adder 194 also receives the output of adder 188 and generates as its output the value of the upsampled target pixel w .

The input pixels and shifting of the multiplexers 174 and 176 are controlled so that the output of adder 188 represents intermediate pixel v_1 (i.e. the intermediate pixel closest to the reference pixel P_R). The input pixels and shifting of the multiplexers 165 to 168, 170 and 172 are controlled so that the output of adder 182 represents either intermediate pixel v_0 or intermediate pixel v_2 depending upon the position of the upsampled target pixel w on the intermediate line segment. The designators V , O_R and O_L input to multiplexers 170 to 176 in Figure 12 refer to the source pixels that are used to interpolate in the vertical, oblique right and oblique left directions respectively.

The linear interpolator 106c constituted by adders 190 and 194 and multiplier 192 performs the calculation representing the non-orthogonal horizontal interpolation between the intermediate pixels.

The following chart sets out the directional interpolations performed by the oblique upsampler 106 for all values of α_x and α_y in order to generate the intermediate pixels v . The value of β for each interpolation direction is also included.

- 18 -

Interpolation Direction	α_y	Intermediate Pixels v_0, v_1, v_2	β
Vertical	$\alpha_y \geq 0$	$v_0 = u_{0,1} + (u_{1,1} - u_{0,1})\alpha_y$ $v_1 = u_{00} + (u_{10} - u_{00})\alpha_y$ $v_2 = u_{01} + (u_{11} - u_{01})\alpha_y$	$\beta = \alpha_x$
	$\alpha_y < 0$	$v_0 = u_{0,1} + (u_{1,1} - u_{0,1})(-\alpha_y)$ $v_1 = u_{00} + (u_{10} - u_{00})(-\alpha_y)$ $v_2 = u_{01} + (u_{11} - u_{01})(-\alpha_y)$	
Oblique Up-Left	$\alpha_y \geq 0$	$v_0 = u_{0,1} + (u_{10} - u_{0,1})\alpha_y$ $v_1 = u_{00} + (u_{11} - u_{00})\alpha_y$ $v_2 = u_{01} + (u_{12} - u_{01})\alpha_y$	$\beta = \alpha_x - \alpha_y$
	$\alpha_y < 0$	$v_0 = u_{0,1} + (u_{1,2} - u_{0,1})(-\alpha_y)$ $v_1 = u_{00} + (u_{1,1} - u_{00})(-\alpha_y)$ $v_2 = u_{01} + (u_{10} - u_{01})(-\alpha_y)$	
Oblique Up-Right	$\alpha_y \geq 0$	$v_0 = u_{0,1} + (u_{1,2} - u_{0,1})\alpha_y$ $v_1 = u_{00} + (u_{1,1} - u_{00})\alpha_y$ $v_2 = u_{01} + (u_{10} - u_{01})\alpha_y$	$\beta = \alpha_x + \alpha_y$
	$\alpha_y < 0$	$v_0 = u_{0,1} + (u_{10} - u_{0,1})(-\alpha_y)$ $v_1 = u_{00} + (u_{11} - u_{00})(-\alpha_y)$ $v_2 = u_{01} + (u_{12} - u_{01})(-\alpha_y)$	

5

As one of skill in the art will appreciate, once the intermediate pixels v and the value of β are known, the upsampled target pixel w can be generated by horizontal interpolation using one of the following equations:

$$w = v_1 + (v_2 - v_1) \beta; \quad \beta \geq 0 \quad (20)$$

10

$$w = v_1 + (v_0 - v_1) (-\beta); \quad \beta < 0 \quad (21)$$

Referring now to Figures 13 to 28, another embodiment of a method for generating upsampled target pixels using seven possible interpolation directions and a four-point interpolator will now be described.

15

Similar to the previous embodiment, the input source data is examined to determine the closest source pixel to the upsampled target pixel w and that source pixel is used as a reference pixel P_R . Source pixels from the two source lines above the upsampled target pixel, and the two source lines below the upsampled target pixel are determined (see block 210 in Figure 23a) and selected pixels of these source lines are compared to determine the interpolation direction to be selected when generating the upsampled target pixel. The source pixels are used to calculate difference values for six possible oblique directions (block 212). The

20

calculations vary depending upon the quadrant of interpolation and are shown in the following chart:

Top-Left	Top-Right	Bottom-Left	Bottom-Right	<u>Associated Interpolation Direction</u>		
$\text{abs}[U_{0,1}-U_{1,0}]$	$\text{abs}[U_{00}-U_{1,1}]$	$\text{abs}[U_{00}-U_{1,1}]$	$\text{abs}[U_{0,1}-U_{1,0}]$	45° line up-right	O_{UR}	(1)
$\text{abs}[U_{00}-U_{1,1}]$	$\text{abs}[U_{0,1}-U_{1,0}]$	$\text{abs}[U_{0,1}-U_{1,0}]$	$\text{abs}[U_{00}-U_{1,1}]$	45° line up-left	O_{UL}	(2)
$\text{abs}[U_{0,1}-U_{1,1}]$	$\text{abs}[U_{00}-U_{1,2}]$	$\text{abs}[U_{00}-U_{1,2}]$	$\text{abs}[U_{0,1}-U_{1,1}]$	26.6° line up-right-right	O_{URR}	(3)
$\text{abs}[U_{00}-U_{1,2}]$	$\text{abs}[U_{0,1}-U_{1,1}]$	$\text{abs}[U_{0,1}-U_{1,1}]$	$\text{abs}[U_{00}-U_{1,2}]$	26.6° line up-left-left	O_{ULL}	(4)
$\text{abs}[U_{1,1}-U_{1,0}]$	$\text{abs}[U_{00}-U_{2,1}]$	$\text{abs}[U_{00}-U_{2,1}]$	$\text{abs}[U_{1,1}-U_{1,0}]$	63.4° line up-up-right	O_{UUR}	(5)
$\text{abs}[U_{00}-U_{2,1}]$	$\text{abs}[U_{1,1}-U_{1,0}]$	$\text{abs}[U_{1,1}-U_{1,0}]$	$\text{abs}[U_{00}-U_{2,1}]$	63.4° line up-up-left	O_{UUL}	(6)
				vertical	V	(7)

15 The interpolation directions O_{UR} , O_{UL} , O_{URR} , O_{ULL} , O_{UUR} , O_{UUL} , and V are illustrated in Figure 13 for the top-left quadrant. No difference values are calculated for the vertical direction V shown.

20 The calculations yield a set of difference values associated with each oblique interpolation direction. The set of difference values to the right of the reference pixel under consideration and the set of difference values to the left of the reference pixel are also used. The three sets of difference values are summed for each oblique interpolation direction yielding a "working" set of difference values. The difference values of this working set are compared with each other to determine the smallest differences within a threshold (blocks 214 and 216). If the smallest differences are sufficiently small to indicate low-frequency edges then they yield a set of possible interpolation directions.

25 The possible interpolation directions are then examined to see if a "boundary crossing" situation occurs (blocks 218 and 220). Figure 14 demonstrates this situation. Here the up-left-left direction representing a white line joining pixels U_{00} and $U_{1,2}$ is being considered, however there is a dark grey "boundary" joining pixels $U_{0,1}$ and $U_{1,1}$. Selecting the up-left-left direction would cross the boundary, and therefore the up-left-left direction can not be considered. This situation is detected by examining differences B_1 and B_2 .

5 The possible interpolation directions are then examined and a single interpolation direction is selected (block 222). Block 222 is expanded in Figure 23b.

- 20 -

If only one oblique interpolation direction is possible then that direction is chosen (blocks 230 and 232). If no low-frequency edges are detected then the vertical interpolation direction is chosen as default (blocks 238 and 240). If both a leftward and a rightward oblique interpolation direction are possible, then the vertical direction is again chosen as default (blocks 248 and 250).

Otherwise only either rightward or leftward oblique interpolation directions are under consideration. If only one of the "extreme" directions (either 26.6° or 63.4° obliques) and the 45° oblique direction are possible then that extreme direction is chosen (blocks 252 and 254). Otherwise both extremes are possible and the "middle" direction (the 45° oblique) is chosen (block 256).

Once the interpolation direction is chosen, the pixels required for the directional interpolation must be determined (block 224). The source data is then directionally interpolated to generate the intermediate pixels (block 226). Finally the intermediate pixels are interpolated horizontally to generate the desired upsampled target pixel w (block 228).

The location of the upsampled target pixel w , is defined by its distances from the reference pixel P_R both in the x and y directions. These distances are represented by the terms α_x and α_y where:

$$-0.5 < \alpha_x \leq 0.5; \text{ and}$$

$$-0.5 < \alpha_y \leq 0.5;$$

α_x being positive to the right and α_y being positive downward.

In this embodiment, a four-point "filter" interpolator is used. Interpolated upsampled target pixels w are calculated as a function of four source pixels, S_0 , S_1 , S_2 and S_3 , and a parameter α representing the distance of the desired pixel from S_1 :

$$w = f(S_0, S_1, S_2, S_3, \alpha)$$

This function may be a four-tap filter with coefficients read from a ROM indexed by parameter α .

Figure 15 shows the case when the vertical interpolation direction V is selected. The target pixel w is positioned in the "Top-Left" quadrant with α_x and α_y

- 21 -

both negative. The intermediate horizontal line segment represented by the values v_i , is determined by vertical four-point interpolation as follows:

$$\begin{aligned}
 v_0 &= f(U_{1,2}, U_{0,2}, U_{-1,2}, U_{-2,2}, -\alpha_y) \\
 v_1 &= f(U_{1,1}, U_{0,1}, U_{-1,1}, U_{-2,1}, -\alpha_y) \\
 5 \quad v_2 &= f(U_{10}, U_{00}, U_{-10}, U_{-20}, -\alpha_y) \\
 v_3 &= f(U_{11}, U_{01}, U_{-11}, U_{-21}, -\alpha_y) \\
 v_4 &= f(U_{12}, U_{02}, U_{-12}, U_{-22}, -\alpha_y)
 \end{aligned}$$

After the five intermediate pixels are calculated, interpolations are performed on these pixels to calculate the value of the desired upsampled target pixel.

10 The distance β , representing the horizontal distance from the target pixel w to the middle intermediate pixel, v_2 must first be calculated. The calculation of β for the other three quadrants is also shown in Figure 15. In the vertical direction case, β is simply equal to α_x for all quadrants. In the case of Figure 15, upsampled target pixel w is in the Top-Left quadrant and β is negative. Thus, the upsampled target pixel is

15 represented by:

$$w = f(v_3, v_2, v_1, v_0, -\beta)$$

In the other quadrants around the reference pixel P_R , the upsampled target pixel w can be similarly calculated.

20 When the oblique up-right direction O_{UR} is selected, the intermediate pixels are interpolated along positively sloped 45° diagonals. Figure 16 shows the up-right case. The target pixel w , is positioned in the Top-Right quadrant with α_x positive and α_y negative. Using four-point interpolation along the indicated diagonals yields:

$$\begin{aligned}
 v_0 &= f(U_{1,3}, U_{0,2}, U_{-1,1}, U_{-2,0}, -\alpha_y) \\
 25 \quad v_1 &= f(U_{1,2}, U_{0,1}, U_{-1,0}, U_{-2,-1}, -\alpha_y) \\
 v_2 &= f(U_{1,1}, U_{0,0}, U_{-1,-1}, U_{-2,-2}, -\alpha_y) \\
 v_3 &= f(U_{10}, U_{01}, U_{-12}, U_{-23}, -\alpha_y) \\
 v_4 &= f(U_{11}, U_{02}, U_{-13}, U_{-24}, -\alpha_y)
 \end{aligned}$$

30 After the five intermediate pixels have been computed non-orthogonal, horizontal interpolation is performed to determine the upsampled target pixel w .

Since the interpolation is on a 45° angle, the horizontal distance from the reference pixel P_R to intermediate pixel v_2 is simply the vertical distance, namely, $-\alpha_y$. Thus, β is simply $\alpha_x + \alpha_y$. In the present case, β is positive so the upsampled target pixel is represented by:

$$w = f(v_1, v_2, v_3, v_4, \beta)$$

For other quadrants, a similar method may be used to determine the upsampled target pixel. The intermediate pixels used and β vary from quadrant to quadrant. Figure 16 also demonstrates the calculation of β for each quadrant. These results are summarized in Table A below.

Interpolating in the oblique up-left direction O_{UL} is very similar to the up-right direction. Intermediate pixels are interpolated along negatively sloped 45° diagonals. In Figure 17, the upsampled target pixel w is in the Bottom-Right quadrant with both α_x and α_y positive. Using four-point interpolation along the diagonals:

$$\begin{aligned} v_0 &= f(U_{-1,3}, U_{0,2}, U_{1,1}, U_{2,0}, \alpha_y) \\ v_1 &= f(U_{-1,2}, U_{0,1}, U_{1,0}, U_{2,-1}, \alpha_y) \\ v_2 &= f(U_{-1,1}, U_{0,0}, U_{1,-1}, U_{2,-2}, \alpha_y) \\ v_3 &= f(U_{-1,0}, U_{0,-1}, U_{1,-2}, U_{2,-3}, \alpha_y) \\ v_4 &= f(U_{-1,-1}, U_{0,-2}, U_{1,-3}, U_{2,-4}, \alpha_y) \end{aligned}$$

The upsampled target pixel is computed by performing non-orthogonal, horizontal interpolation on the intermediate pixels. Again, since the interpolation is along a 45° diagonal, the horizontal distances to the intermediate pixels are equal to the vertical distances. For the Bottom-Right quadrant, $\beta = \alpha_x - \alpha_y$. With β positive, the upsampled target pixel is represented by:

$$w = f(v_1, v_2, v_3, v_4, \beta)$$

Again a similar method applies for the other quadrants with β varying from quadrant to quadrant. Figure 17 demonstrates the calculation of β for the other quadrants with the results in Table A.

When the oblique up-right-right (O_{URR}) or up-left-left (O_{ULL}) interpolation directions are selected, interpolation is along the 26.6° diagonals. The source pixels/diagonals used for four-point interpolation are shown for both the up-right-right and up-left-left directions in Figure 18.

Figure 19 shows some of the source pixels used in up-right-right O_{URR} interpolation for an upsampled target pixel w in the Bottom-Left quadrant, with α_x negative and α_y positive. Using four-point interpolation along the positively sloped 26.6° diagonals:

$$\begin{aligned} v_0 &= f(U_{-10}, U_{0-2}, U_{1-4}, U_{2-6}, \alpha_y) \\ v_1 &= f(U_{-11}, U_{0-1}, U_{1-3}, U_{2-5}, \alpha_y) \\ v_2 &= f(U_{-12}, U_{00}, U_{1-2}, U_{2-4}, \alpha_y) \\ v_3 &= f(U_{-13}, U_{01}, U_{1-1}, U_{2-3}, \alpha_y) \\ v_4 &= f(U_{-14}, U_{02}, U_{10}, U_{2-2}, \alpha_y) \end{aligned}$$

The upsampled target pixel is calculated by horizontal interpolation. Since interpolation is along the 26.6° diagonals, horizontal distances to the intermediate pixels are twice the vertical distance. For the Bottom-Left quadrant $\beta = 2\alpha_y - (-\alpha_x) = \alpha_x + 2\alpha_y$. With β positive, the upsampled target pixel is represented by:

$$w = f(v_1, v_2, v_3, v_4, \beta)$$

Similarly for other quadrants. Figure 19 shows the calculation of β for other quadrants as summarized in Table A.

When the up-left-left interpolation direction O_{ULL} is selected for interpolation, intermediate pixels are interpolated along negatively sloped 26.6° diagonals. Some of the source pixels used are shown in Figure 20 for the Top-Left quadrant with both α_x and α_y negative. Using four-point interpolation:

$$\begin{aligned} v_0 &= f(U_{10}, U_{0-2}, U_{-1-4}, U_{-2-6}, -\alpha_y) \\ v_1 &= f(U_{11}, U_{0-1}, U_{-1-3}, U_{-2-5}, -\alpha_y) \\ v_2 &= f(U_{12}, U_{00}, U_{-1-2}, U_{-2-4}, -\alpha_y) \\ v_3 &= f(U_{13}, U_{01}, U_{-1-1}, U_{-2-3}, -\alpha_y) \\ v_4 &= f(U_{14}, U_{02}, U_{-10}, U_{-2-2}, -\alpha_y) \end{aligned}$$

Again since interpolation is along 26.6° diagonals, the horizontal distance to the intermediate pixels is equal to twice the vertical. For the Top-Left quadrant $\beta = -2\alpha_y - (-\alpha_x) = \alpha_x - 2\alpha_y$. With β positive, the upsampled target pixel is represented by:

$$w = f(v_1, v_2, v_3, v_4, \beta)$$

A similar method applies for the other quadrant with varying β . Figure 20 demonstrates the calculation of β for other quadrants. The results are summarized in Table A.

When either the oblique up-up-right (O_{UUR}) or up-up-left (O_{UUL}) interpolation directions are selected, the directional interpolation involves interpolating across more source lines than in the previous cases. To avoid requiring too many source lines, linear interpolation is used for directional interpolation for the up-up-right and up-up-left directions.

Figure 21 illustrates the interpolations required for the up-up-right, O_{UUR} , interpolation direction. The upsampled target pixel w , is in the Bottom-Right quadrant with both α_x and α_y positive. Interpolation is along positively sloped, 63.4° diagonals. There are two types of these offsetting diagonals, namely the "inside" and "outside" diagonals as represented by the lines $U_{1,1}U_{10}$ and $U_{00}U_{2,1}$ respectively. The linear upsampling parameter for the outside diagonal is given by the vertical distance α_y from the intermediate pixel to the source pixel. Since interpolation skips a source line, the linear upsampling parameter is halved, i.e. $\alpha_y/2$. Similarly for the inside diagonal, the parameter is $(1-\alpha_y)/2$. Using linear interpolation, the intermediate pixels are calculated as follows:

$$v_0 = (1-0.5(1-\alpha_y))U_{1,1} + 0.5(1-\alpha_y)U_{10}$$

$$v_1 = (1-0.5\alpha_y)U_{00} + 0.5\alpha_yU_{2,1}$$

$$v_2 = (1-0.5(1-\alpha_y))U_{10} + 0.5(1-\alpha_y)U_{11}$$

$$v_3 = (1-0.5\alpha_y)U_{01} + 0.5\alpha_yU_{20}$$

$$v_4 = (1-0.5(1-\alpha_y))U_{11} + 0.5(1-\alpha_y)U_{12}$$

The upsampled target pixel is calculated by non-orthogonal horizontal interpolation. Four-point interpolation is still used horizontally. Since interpolation is along the 63.4° diagonals, horizontal distances to the intermediate pixels are half the vertical distance. Additionally, since the spacing between intermediate pixels is half the spacing for the previous interpolation directions, β is twice the distance from the centre intermediate pixel to the target pixel. For the Bottom-Right quadrant $0.5 = \alpha_x - 0.5\beta + 0.5\alpha_y$. So $\beta = -1 + 2\alpha_x + \alpha_y$. With β negative, the upsampled target pixel is represented by:

- 25 -

$$w = f(v_3, v_2, v_1, v_0, -\beta)$$

Similarly for other quadrants. Figure 21 shows the calculation of β for other quadrants as summarized in Table A.

For the oblique up-up-left O_{UUL} , interpolation direction shown in Figure 22 the upsampled target pixel w , is in the Top-Right quadrant with α_x positive and α_y negative. Interpolation is along the negatively sloped 63.4° diagonals. The linear upsampling parameter for the inside diagonal is $(1 + \alpha_y)/2$ and the parameter for the outside diagonals is $-\alpha_y/2$. Once again using linear interpolation:

$$v_0 = (1 - 0.5(1 + \alpha_y))U_{-1,-1} + 0.5(1 + \alpha_y)U_{10}$$

$$v_1 = (1 - 0.5(-\alpha_y))U_{00} + 0.5(-\alpha_y)U_{-2,-1}$$

$$v_2 = (1 - 0.5(1 + \alpha_y))U_{-10} + 0.5(1 + \alpha_y)U_{11}$$

$$v_3 = (1 - 0.5(-\alpha_y))U_{01} + 0.5(-\alpha_y)U_{-20}$$

$$v_4 = (1 - 0.5(1 + \alpha_y))U_{-11} + 0.5(1 + \alpha_y)U_{12}$$

Four-point horizontal interpolation is performed on the intermediate pixels. Once again the horizontal distances to the intermediate pixels are half the vertical distances, and β is twice the distance from the centre intermediate pixel to the target pixel. For the case of the upsampled target pixel w being in the Top-Right quadrant, $\beta = 2(\alpha_x + (-0.5\alpha_y) - 1) = -1 + 2\alpha_x - \alpha_y$. Since β is positive, the upsampled target pixel is represented by:

$$w = f(v_1, v_2, v_3, v_4, \beta)$$

A similar method applies for the other quadrants. Figure 22 shows the calculation of β for other quadrants as summarized in Table A.

Figure 24 shows a basic source data interpolation apparatus 300 for performing the above method. As can be seen, apparatus 300 is the same as the apparatus used in the first embodiment. The same block diagram applies to the second embodiment just described. Source data D_{in} is passed through a delay circuit 302 which extracts the necessary source pixels required for interpolation. These source pixels are used by the decision circuit 304 to select an interpolation direction. The selected interpolation direction is passed to the β calculator 308 to determine the value of β . The source pixels, interpolation direction, and β are all passed to the

- 26 -

oblique upsampler 306 for directional interpolation to generate the upsampled target pixel w.

Figure 25 better illustrates the delay circuit 302 of the present embodiment. It includes three line stores 344, 370, and 396 which give access to four consecutive source lines and forty-eight delay elements 320 to 342, 346 to 368, 372 to 394, and 398 to 420 which in turn give access to thirteen adjacent source pixels on each of the four source lines. The delay circuit extracts the 13x4 source matrix needed for interpolation. When interpolating in the upper quadrants, the pixels extracted correspond to the source pixels labelled.

The decision circuit 304 is better illustrated in Figure 26. Input source pixels are fed to six subtractors 446, 452, 458, 464, 470, and 476. Each subtractor calculates a difference value corresponding to one of the six possible oblique directions. Two delay elements for each difference value 448, 450, 454, 456, 460, 462, 466, 468, 472, 474, 478, and 480 provide access to three consecutive difference values for each direction. These difference values are passed to a selection means 482 which selects either one of the six possible oblique interpolation directions or the vertical interpolation direction. The source pixels labelled correspond to interpolation in the top-left quadrant.

The β calculator shown in Figure 27 comprises a three input adder 508. The adder inputs are the outputs of three multiplexers 502, 504, and 506. The output of multiplexer 502 is 0, 1, or -1 depending on the output of decoder 500 which takes the interpolation direction selected and the quadrant of the upsampled target pixel. Multiplexer 504 outputs either α_x or $2\alpha_x$ depending on the interpolation direction selected. Finally multiplexer 506 outputs either α_y , $-\alpha_y$, $2\alpha_y$, or $-2\alpha_y$ depending also on the interpolation direction. The output of adder 508 is β .

The oblique upsampler 306 is best illustrated in Figure 28. It comprises three main parts; namely an input decoder 532, a directional interpolator 306a and a horizontal interpolator 306b. The input source pixels are decoded in the input decoder 532 according to the quadrant in which the upsampled target pixel is located and the interpolation direction. The required source pixels are sent to the directional interpolator 306a which performs four 4-point interpolations to generate four intermediate pixels. The 4-point directional interpolation is performed by four 4-

- 27 -

tap filters 534, 536, 538, and 540 whose coefficients are supplied from a ROM, 530, addressed by $|\alpha_y|$. The four intermediate pixels are passed to the 4-point horizontal interpolator 306b to calculate the upsampled target pixel w . The horizontal interpolator 306b is composed of a 4-tap filter 544 and a ROM, 542, addressed by β .

5

Table A below outlines the directional interpolations performed to generate the intermediate pixels for each of the seven interpolation directions. The Table also includes the formula for calculating β for the various directions and quadrants.

TABLE A

Interpolation Direction	Quadrant	Intermediate Pixels v_0, v_1, v_2, v_3, v_4	β
Vertical V	Top-Left $\alpha_x < 0, \alpha_y < 0$	$v_0 = f(U_{-1,2}, U_{0,2}, U_{-1,2}, U_{-2,2}, -\alpha_y)$ $v_1 = f(U_{-1,1}, U_{0,1}, U_{-1,1}, U_{-2,1}, -\alpha_y)$ $v_2 = f(U_{-1,0}, U_{0,0}, U_{-1,0}, U_{-2,0}, -\alpha_y)$ $v_3 = f(U_{-1,1}, U_{0,1}, U_{-1,1}, U_{-2,1}, -\alpha_y)$ $v_4 = f(U_{-1,2}, U_{0,2}, U_{-1,2}, U_{-2,2}, -\alpha_y)$	α_x
	Top-Right $\alpha_x \geq 0, \alpha_y < 0$	$v_0 = f(U_{1,2}, U_{0,2}, U_{1,2}, U_{2,2}, -\alpha_y)$ $v_1 = f(U_{1,1}, U_{0,1}, U_{1,1}, U_{2,1}, -\alpha_y)$ $v_2 = f(U_{1,0}, U_{0,0}, U_{1,0}, U_{2,0}, -\alpha_y)$ $v_3 = f(U_{1,1}, U_{0,1}, U_{1,1}, U_{2,1}, -\alpha_y)$ $v_4 = f(U_{1,2}, U_{0,2}, U_{1,2}, U_{2,2}, -\alpha_y)$	α_x
	Bottom-Left $\alpha_x < 0, \alpha_y \geq 0$	$v_0 = f(U_{-1,2}, U_{0,2}, U_{-1,2}, U_{-2,2}, \alpha_y)$ $v_1 = f(U_{-1,1}, U_{0,1}, U_{-1,1}, U_{-2,1}, \alpha_y)$ $v_2 = f(U_{-1,0}, U_{0,0}, U_{-1,0}, U_{-2,0}, \alpha_y)$ $v_3 = f(U_{-1,1}, U_{0,1}, U_{-1,1}, U_{-2,1}, \alpha_y)$ $v_4 = f(U_{-1,2}, U_{0,2}, U_{-1,2}, U_{-2,2}, \alpha_y)$	α_x
	Bottom-Right $\alpha_x \geq 0, \alpha_y \geq 0$	$v_0 = f(U_{1,2}, U_{0,2}, U_{1,2}, U_{2,2}, \alpha_y)$ $v_1 = f(U_{1,1}, U_{0,1}, U_{1,1}, U_{2,1}, \alpha_y)$ $v_2 = f(U_{1,0}, U_{0,0}, U_{1,0}, U_{2,0}, \alpha_y)$ $v_3 = f(U_{1,1}, U_{0,1}, U_{1,1}, U_{2,1}, \alpha_y)$ $v_4 = f(U_{1,2}, U_{0,2}, U_{1,2}, U_{2,2}, \alpha_y)$	α_x

Up-Right O_{UR}	Top-Left $\alpha_x < 0, \alpha_y < 0$	$v_0 = f(U_{1,4}, U_{0,3}, U_{1,2}, U_{2,1}, -\alpha_y)$ $v_1 = f(U_{1,3}, U_{0,2}, U_{1,1}, U_{2,0}, -\alpha_y)$ $v_2 = f(U_{1,2}, U_{0,1}, U_{1,0}, U_{2,1}, -\alpha_y)$ $v_3 = f(U_{1,1}, U_{0,0}, U_{1,1}, U_{2,2}, -\alpha_y)$ $v_4 = f(U_{1,0}, U_{0,1}, U_{1,2}, U_{2,3}, -\alpha_y)$	$1 + \alpha_x + \alpha_y$ $\alpha_x + \alpha_y$
	Top-Right $\alpha_x \geq 0, \alpha_y < 0$	$v_0 = f(U_{1,3}, U_{0,2}, U_{1,1}, U_{2,0}, -\alpha_y)$ $v_1 = f(U_{1,2}, U_{0,1}, U_{1,0}, U_{2,1}, -\alpha_y)$ $v_2 = f(U_{1,1}, U_{0,0}, U_{1,1}, U_{2,2}, -\alpha_y)$ $v_3 = f(U_{1,0}, U_{0,1}, U_{1,2}, U_{2,3}, -\alpha_y)$ $v_4 = f(U_{1,1}, U_{0,2}, U_{1,3}, U_{2,4}, -\alpha_y)$	$\alpha_x + \alpha_y$
	Bottom-Left $\alpha_x < 0, \alpha_y \geq 0$	$v_0 = f(U_{1,1}, U_{0,2}, U_{1,3}, U_{2,4}, \alpha_y)$ $v_1 = f(U_{1,0}, U_{0,1}, U_{1,2}, U_{2,3}, \alpha_y)$ $v_2 = f(U_{1,1}, U_{0,0}, U_{1,1}, U_{2,2}, \alpha_y)$ $v_3 = f(U_{1,2}, U_{0,1}, U_{1,0}, U_{2,1}, \alpha_y)$ $v_4 = f(U_{1,3}, U_{0,2}, U_{1,1}, U_{2,0}, \alpha_y)$	$-$ $1 + \alpha_x + \alpha_y$
	Bottom-Right $\alpha_x \geq 0, \alpha_y \geq 0$	$v_0 = f(U_{1,0}, U_{0,1}, U_{1,2}, U_{2,3}, \alpha_y)$ $v_1 = f(U_{1,1}, U_{0,0}, U_{1,1}, U_{2,2}, \alpha_y)$ $v_2 = f(U_{1,2}, U_{0,1}, U_{1,0}, U_{2,1}, \alpha_y)$ $v_3 = f(U_{1,3}, U_{0,2}, U_{1,1}, U_{2,0}, \alpha_y)$ $v_4 = f(U_{1,4}, U_{0,3}, U_{1,2}, U_{2,1}, \alpha_y)$	

Up-Left O_{UL}	Top-Left $\alpha_x < 0, \alpha_y < 0$	$v_0 = f(U_{-1,-1}, U_{0,-2}, U_{-1,-3}, U_{-2,-4}, -\alpha_y)$ $v_1 = f(U_{10}, U_{0,-1}, U_{-1,-2}, U_{-2,-3}, -\alpha_y)$ $v_2 = f(U_{11}, U_{00}, U_{-1,-1}, U_{-2,-2}, -\alpha_y)$ $v_3 = f(U_{12}, U_{01}, U_{-10}, U_{-2,-1}, -\alpha_y)$ $v_4 = f(U_{13}, U_{02}, U_{-11}, U_{-20}, -\alpha_y)$	$\alpha_x - \alpha_y$ $-1 + \alpha_x - \alpha_y$
	Top-Right $\alpha_x \geq 0, \alpha_y < 0$	$v_0 = f(U_{10}, U_{0,-1}, U_{-1,-2}, U_{-2,-3}, -\alpha_y)$ $v_1 = f(U_{11}, U_{00}, U_{-1,-1}, U_{-2,-2}, -\alpha_y)$ $v_2 = f(U_{12}, U_{01}, U_{-10}, U_{-2,-1}, -\alpha_y)$ $v_3 = f(U_{13}, U_{02}, U_{-11}, U_{-20}, -\alpha_y)$ $v_4 = f(U_{14}, U_{03}, U_{-12}, U_{-21}, -\alpha_y)$	$1 + \alpha_x - \alpha_y$
	Bottom-Left $\alpha_x < 0, \alpha_y \geq 0$	$v_0 = f(U_{-1,-4}, U_{0,-3}, U_{1,-2}, U_{2,-1}, \alpha_y)$ $v_1 = f(U_{-1,-3}, U_{0,-2}, U_{1,-1}, U_{20}, \alpha_y)$ $v_2 = f(U_{-1,-2}, U_{0,-1}, U_{10}, U_{21}, \alpha_y)$ $v_3 = f(U_{-1,-1}, U_{00}, U_{11}, U_{22}, \alpha_y)$ $v_4 = f(U_{-10}, U_{01}, U_{12}, U_{23}, \alpha_y)$	$\alpha_x - \alpha_y$
	Bottom-Right $\alpha_x \geq 0, \alpha_y \geq 0$	$v_0 = f(U_{-1,-3}, U_{0,-2}, U_{1,-1}, U_{20}, \alpha_y)$ $v_1 = f(U_{-1,-2}, U_{0,-1}, U_{10}, U_{21}, \alpha_y)$ $v_2 = f(U_{-1,-1}, U_{00}, U_{11}, U_{22}, \alpha_y)$ $v_3 = f(U_{-10}, U_{01}, U_{12}, U_{23}, \alpha_y)$ $v_4 = f(U_{-11}, U_{02}, U_{13}, U_{24}, \alpha_y)$	

Up-Right-Right O_{URR}	Top-Left $\alpha_x < 0, \alpha_y < 0$	$v_0 = f(U_{1,5}, U_{0,3}, U_{1,1}, U_{2,1}, -\alpha_y)$ $v_1 = f(U_{1,4}, U_{0,2}, U_{1,0}, U_{2,2}, -\alpha_y)$ $v_2 = f(U_{1,3}, U_{0,1}, U_{1,1}, U_{2,3}, -\alpha_y)$ $v_3 = f(U_{1,2}, U_{0,0}, U_{1,2}, U_{2,4}, -\alpha_y)$ $v_4 = f(U_{1,1}, U_{0,1}, U_{1,3}, U_{2,5}, -\alpha_y)$	$1 + \alpha_x + 2\alpha_y$
	Top-Right $\alpha_x \geq 0, \alpha_y < 0$	$v_0 = f(U_{1,4}, U_{0,2}, U_{1,0}, U_{2,2}, -\alpha_y)$ $v_1 = f(U_{1,3}, U_{0,1}, U_{1,1}, U_{2,3}, -\alpha_y)$ $v_2 = f(U_{1,2}, U_{0,0}, U_{1,2}, U_{2,4}, -\alpha_y)$ $v_3 = f(U_{1,1}, U_{0,1}, U_{1,3}, U_{2,5}, -\alpha_y)$ $v_4 = f(U_{1,0}, U_{0,2}, U_{1,4}, U_{2,6}, -\alpha_y)$	$\alpha_x + 2\alpha_y$ $\alpha_x + 2\alpha_y$
	Bottom-Left $\alpha_x < 0, \alpha_y \geq 0$	$v_0 = f(U_{1,0}, U_{0,2}, U_{1,4}, U_{2,6}, \alpha_y)$ $v_1 = f(U_{1,1}, U_{0,1}, U_{1,3}, U_{2,5}, \alpha_y)$ $v_2 = f(U_{1,2}, U_{0,0}, U_{1,2}, U_{2,4}, \alpha_y)$ $v_3 = f(U_{1,3}, U_{0,1}, U_{1,1}, U_{2,3}, \alpha_y)$ $v_4 = f(U_{1,4}, U_{0,2}, U_{1,0}, U_{2,2}, \alpha_y)$	 $-$ $1 + \alpha_x + 2\alpha_y$
	Bottom-Right $\alpha_x \geq 0, \alpha_y \geq 0$	$v_0 = f(U_{1,1}, U_{0,1}, U_{1,3}, U_{2,5}, \alpha_y)$ $v_1 = f(U_{1,2}, U_{0,0}, U_{1,2}, U_{2,4}, \alpha_y)$ $v_2 = f(U_{1,3}, U_{0,1}, U_{1,1}, U_{2,3}, \alpha_y)$ $v_3 = f(U_{1,4}, U_{0,2}, U_{1,0}, U_{2,2}, \alpha_y)$ $v_4 = f(U_{1,5}, U_{0,3}, U_{1,1}, U_{2,1}, \alpha_y)$	

Up-Left-Left O_{ULL}	Top-Left $\alpha_x < 0, \alpha_y < 0$	$v_0 = f(U_{10}, U_{0-2}, U_{-1-4}, U_{-2-6}, -\alpha_y)$ $v_1 = f(U_{11}, U_{0-1}, U_{-1-3}, U_{-2-5}, -\alpha_y)$ $v_2 = f(U_{12}, U_{00}, U_{-1-2}, U_{-2-4}, -\alpha_y)$ $v_3 = f(U_{13}, U_{01}, U_{-1-1}, U_{-2-3}, -\alpha_y)$ $v_4 = f(U_{14}, U_{02}, U_{-10}, U_{-2-2}, -\alpha_y)$	$\alpha_x - 2\alpha_y$ $-1 + \alpha_x - 2\alpha_y$
	Top-Right $\alpha_x \geq 0, \alpha_y < 0$	$v_0 = f(U_{11}, U_{0-1}, U_{-1-3}, U_{-2-5}, -\alpha_y)$ $v_1 = f(U_{12}, U_{00}, U_{-1-2}, U_{-2-4}, -\alpha_y)$ $v_2 = f(U_{13}, U_{01}, U_{-1-1}, U_{-2-3}, -\alpha_y)$ $v_3 = f(U_{14}, U_{02}, U_{-10}, U_{-2-2}, -\alpha_y)$ $v_4 = f(U_{15}, U_{03}, U_{-11}, U_{-2-1}, -\alpha_y)$	$1 + \alpha_x - 2\alpha_y$
	Bottom-Left $\alpha_x < 0, \alpha_y \geq 0$	$v_0 = f(U_{-1-5}, U_{0-3}, U_{1-1}, U_{21}, \alpha_y)$ $v_1 = f(U_{-1-4}, U_{0-2}, U_{10}, U_{22}, \alpha_y)$ $v_2 = f(U_{-1-3}, U_{0-1}, U_{11}, U_{23}, \alpha_y)$ $v_3 = f(U_{-1-2}, U_{00}, U_{12}, U_{24}, \alpha_y)$ $v_4 = f(U_{-1-1}, U_{01}, U_{13}, U_{25}, \alpha_y)$	$\alpha_x - 2\alpha_y$
	Bottom-Right $\alpha_x \geq 0, \alpha_y \geq 0$	$v_0 = f(U_{-1-4}, U_{0-2}, U_{10}, U_{22}, \alpha_y)$ $v_1 = f(U_{-1-3}, U_{0-1}, U_{11}, U_{23}, \alpha_y)$ $v_2 = f(U_{-1-2}, U_{00}, U_{12}, U_{24}, \alpha_y)$ $v_3 = f(U_{-1-1}, U_{01}, U_{13}, U_{25}, \alpha_y)$ $v_4 = f(U_{-10}, U_{02}, U_{14}, U_{26}, \alpha_y)$	

Up-Up-Right O_{UUR}	Top-Left $\alpha_x < 0, \alpha_y < 0$	$v_0 = U_{-1,-1} + (U_{1,-2} - U_{-1,-1})(0.5+0.5\alpha_y)$ $v_1 = U_{0,-1} + (U_{-2,0} - U_{0,-1})(-0.5\alpha_y)$ $v_2 = U_{-1,0} + (U_{1,-1} - U_{-1,0})(0.5+0.5\alpha_y)$ $v_3 = U_{0,0} + (U_{-2,1} - U_{0,0})(-0.5\alpha_y)$ $v_4 = U_{-1,1} + (U_{1,0} - U_{-1,1})(0.5+0.5\alpha_y)$	$1+2\alpha_x + \alpha_y$
	Top-Right $\alpha_x \geq 0, \alpha_y < 0$	$v_0 = U_{0,-1} + (U_{-2,0} - U_{0,-1})(-0.5\alpha_y)$ $v_1 = U_{-1,0} + (U_{1,-1} - U_{-1,0})(0.5+0.5\alpha_y)$ $v_2 = U_{0,0} + (U_{-2,1} - U_{0,0})(-0.5\alpha_y)$ $v_3 = U_{-1,1} + (U_{1,0} - U_{-1,1})(0.5+0.5\alpha_y)$ $v_4 = U_{0,1} + (U_{-2,2} - U_{0,1})(-0.5\alpha_y)$	$2\alpha_x + \alpha_y$
	Bottom-Left $\alpha_x < 0, \alpha_y \geq 0$	$v_0 = U_{0,-1} + (U_{2,-2} - U_{0,-1})(0.5\alpha_y)$ $v_1 = U_{1,-1} + (U_{-1,0} - U_{1,-1})(0.5-0.5\alpha_y)$ $v_2 = U_{0,0} + (U_{2,-1} - U_{0,0})(0.5\alpha_y)$ $v_3 = U_{1,0} + (U_{-1,1} - U_{1,0})(0.5-0.5\alpha_y)$ $v_4 = U_{0,1} + (U_{2,0} - U_{0,1})(0.5\alpha_y)$	$2\alpha_x + \alpha_y$
	Bottom-Right $\alpha_x \geq 0, \alpha_y \geq 0$	$v_0 = U_{1,-1} + (U_{-1,0} - U_{1,-1})(0.5-0.5\alpha_y)$ $v_1 = U_{0,0} + (U_{2,-1} - U_{0,0})(0.5\alpha_y)$ $v_2 = U_{1,0} + (U_{-1,1} - U_{1,0})(0.5-0.5\alpha_y)$ $v_3 = U_{0,1} + (U_{2,0} - U_{0,1})(0.5\alpha_y)$ $v_4 = U_{1,1} + (U_{-1,2} - U_{1,1})(0.5-0.5\alpha_y)$	- $1+2\alpha_x + \alpha_y$

Up-Up-Left O_{UUL}	Top-Left $\alpha_x < 0, \alpha_y < 0$	$v_0 = U_{0-1} + (U_{-2-2} - U_{0-1})(-0.5\alpha_y)$ $v_1 = U_{-1-1} + (U_{10} - U_{-1-1})(0.5+0.5\alpha_y)$ $v_2 = U_{00} + (U_{-2-1} - U_{00})(-0.5\alpha_y)$ $v_3 = U_{-10} + (U_{11} - U_{-10})(0.5+0.5\alpha_y)$ $v_4 = U_{01} + (U_{-20} - U_{01})(-0.5\alpha_y)$	$2\alpha_x - \alpha_y$
	Top-Right $\alpha_x \geq 0, \alpha_y < 0$	$v_0 = U_{-1-1} + (U_{10} - U_{-1-1})(0.5+0.5\alpha_y)$ $v_1 = U_{00} + (U_{-2-1} - U_{00})(-0.5\alpha_y)$ $v_2 = U_{-10} + (U_{11} - U_{-10})(0.5+0.5\alpha_y)$ $v_3 = U_{01} + (U_{-20} - U_{01})(-0.5\alpha_y)$ $v_4 = U_{-11} + (U_{12} - U_{-11})(0.5+0.5\alpha_y)$	$-1+2\alpha_x - \alpha_y$
	Bottom-Left $\alpha_x < 0, \alpha_y \geq 0$	$v_0 = U_{-1-1} + (U_{-1-2} - U_{-1-1})(0.5-0.5\alpha_y)$ $v_1 = U_{0-1} + (U_{20} - U_{0-1})(0.5\alpha_y)$ $v_2 = U_{10} + (U_{-1-1} - U_{10})(0.5-0.5\alpha_y)$ $v_3 = U_{00} + (U_{21} - U_{00})(0.5\alpha_y)$ $v_4 = U_{-1-1} + (U_{-10} - U_{-1-1})(0.5-0.5\alpha_y)$	$1+2\alpha_x - \alpha_y$
	Bottom-Right $\alpha_x \geq 0, \alpha_y \geq 0$	$v_0 = U_{0-1} + (U_{20} - U_{0-1})(0.5\alpha_y)$ $v_1 = U_{10} + (U_{-1-1} - U_{10})(0.5-0.5\alpha_y)$ $v_2 = U_{00} + (U_{21} - U_{00})(0.5\alpha_y)$ $v_3 = U_{-1-1} + (U_{-10} - U_{-1-1})(0.5-0.5\alpha_y)$ $v_4 = U_{01} + (U_{22} - U_{01})(0.5\alpha_y)$	$2\alpha_x - \alpha_y$

15 As mentioned previously, once the intermediate pixels are generated four-point horizontal interpolation is performed on the intermediate pixels to generate the desired upsampled target pixel w.

$$w = f(v_1, v_2, v_3, v_4, \beta) \quad \beta \geq 0$$

$$w = f(v_3, v_2, v_1, v_0, -\beta) \quad \beta < 0$$

20 The present methods and apparatuses allow upsampled target pixels to be generated so that when an image of the source data is to be enlarged and recreated, the resolution of the enlarged image is high and stairstepping is minimized.

As should be apparent to those of skill in the art, various modifications may be made to the present invention without departing from its scope as defined by the appended claims. For example, although two embodiments of the invention have been described herein, it should be apparent that other embodiments are possible and that the number of oblique directions in each set can be chosen arbitrarily.

WHAT IS CLAIMED IS:

1. A method of generating an upsampled target pixel from input source data comprising the steps of:

5 (i) comparing pixels of different lines of said source data in a region surrounding an upsampled target pixel to be generated in at least two different directions;

(ii) selecting an interpolation direction based on the comparisons of step (i);

10 (iii) interpolating between selected pixels of different lines of said source data in the interpolation direction determined at step (ii) and computing intermediate pixels on a line segment passing through said upsampled target pixel; and

(iv) interpolating between the intermediate pixels to generate said upsampled target pixel.

15 2. The method of claim 1 wherein said line segment is parallel to the lines of said source data.

3. The method of claim 2 wherein at step (i), pixels of different lines of said source data are compared in at least three different directions.

20 4. The method of claim 3 wherein at step (i) pixels of different lines of said source data are compared in only three different directions, said three different directions including a vertical direction, and opposite oblique directions forming an angle with said vertical direction, said angle being in the range $0 < \text{angle} < 90^\circ$, the pixels of said source data being compared in each of said oblique directions being located on opposite sides of said upsampled target pixel.

25 5. The method of claim 4 wherein during the comparisons of step (i), difference values between the compared pixels are generated, the difference values being used at step (ii) to select the interpolation direction.

6. The method of claim 5 wherein prior to step (ii), the difference values resulting from the comparisons in said three directions are compared to determine whether the differences therebetween are within a threshold, the interpolation direction associated with the smallest difference value being selected when none of the differences are within the threshold, the selection of said interpolation direction at step (ii) being influenced when the difference between at least two difference values is within the threshold.

7. The method of claim 6 wherein at step (ii), when the differences between the difference values resulting from the comparisons in all three directions are within said threshold, the vertical interpolation direction is selected at step (ii).

8. The method of claim 7 wherein when the difference between the difference values resulting from the comparisons in the vertical direction and only one of the oblique directions is within the threshold, that oblique interpolation direction is selected at step (ii).

9. The method of claim 8 wherein when the difference between the difference values resulting from the comparisons in the two oblique directions is within the threshold, additional pixels of the different lines of source data are compared in the opposite oblique directions, the pixels of said source data being compared in each of said oblique directions being located on opposite sides of said upsampled target pixel, the difference values generated as a result of these comparisons being compared to influence the selection of said interpolation direction at step (ii).

10. The method of claim 9 wherein when the difference between the difference values resulting from the comparisons of the additional pixels is within the threshold, the vertical interpolation direction is selected at step (ii).

11. The method of claim 4 wherein during step (i), the pixels of the different lines of source data are examined to determine the closest pixel to said

- 36 -

upsampled target pixel and wherein said region is defined by consecutive pixels in two lines of source data that form the corners of a rectangular region surrounding said upsampled target pixel, one of the pixels of which is said closest pixel, said oblique directions forming a 45° angle with said vertical direction and constituting diagonals of said rectangular region.

12. The method of claim 11 wherein during the comparisons of step (i), difference values between the compared pixels are generated, the difference values being used at step (ii) to select the interpolation direction.

13. The method of claim 12 wherein prior to step (ii), the difference values resulting from the comparisons in said three directions are compared to determine whether the differences therebetween are within a threshold, the interpolation direction associated with the smallest difference value being selected when none of the differences are within the threshold, the selection of said interpolation direction at step (ii) being influenced when the difference between at least two difference values is within the threshold.

14. The method of claim 13 wherein at step (ii), when the differences between the difference values resulting from the comparisons in all three interpolation directions are within said threshold, the vertical interpolation direction is selected at step (ii).

15. The method of claim 14 wherein when the difference between the difference values resulting from the comparisons in the vertical direction and only one of the oblique directions is within the threshold, that oblique interpolation direction is selected at step (ii).

16. The method of claim 15 wherein when the difference between the difference values resulting from the comparisons in the two oblique directions is within the threshold, additional pixels of the different lines of source data are compared in the opposite oblique directions, the difference values generated as a

result of these comparisons being compared to influence the selection of said interpolation direction at step (ii).

17. The method of claim 16 wherein the difference between the difference values resulting from the comparisons of the additional pixels is within the threshold,
5 the vertical interpolation direction is selected at step (ii).

18. The method of claim 3 wherein at step (i), pixels of different lines of said source data are compared in at least seven different directions.

19. The method of claim 18 wherein at step (i) pixels of said source data are compared in only seven different directions, said seven different directions
10 including a vertical direction and two sets of opposite oblique directions forming an angle with said vertical direction, said angle being in the range of $0 < \text{angle} < 90^\circ$, each set of opposite oblique directions including three oblique directions, each forming a different angle with said vertical direction, the pixels of the source data being compared in each set of oblique directions, being located on opposite sides of said
15 upsampled target pixel.

20. The method of claim 19 wherein one of the oblique directions in each set forms a 26.6° angle with the vertical direction, another of the oblique direction in each set forms a 63.4° angle with the vertical direction and another of the oblique directions in each set forms a 45° angle with said vertical direction.

20 21. The method of claim 20 wherein during the comparisons of step (i), difference values between the compared pixels are generated, the difference values being used at step (ii) to select the interpolation direction.

22. The method of claim 21 wherein the difference values generated during comparisons in said oblique directions in each set are summed to yield a
25 working difference value, the working difference values and the difference value associated with said vertical direction being compared to determine whether the

- 38 -

5 differences therebetween are within a threshold, the interpolation directions associated with the difference values within the threshold yielding a set of possible interpolation directions, the interpolation directions in said set being initially examined to eliminate interpolation directions which interpolate between pixels in a direction that cross a boundary, the remaining interpolation directions being compared further at step (ii) to select said interpolation direction.

10 23. The method of claim 22 wherein after interpolation directions which cross a boundary have been eliminated, at step (ii) if only one oblique interpolation direction remains in said set then that interpolation direction is selected, if opposite oblique interpolation directions remain in said set then said vertical interpolation is selected and if more than one oblique interpolation direction in the same direction exists in the set then the 45° oblique interpolation direction is selected.

15 24. The method of claim 23 wherein at step (iii) interpolations between selected pixels on four different lines of said source data are made in the selected interpolation direction to yield five intermediate pixels.

25. An apparatus for generating an upsampled target pixel from input source data comprising:
comparison means for comparing pixels of different lines of said source data in a region surrounding an upsampled target pixel, in at least two different directions;
20 selection means for selecting an interpolation direction in response to said comparison means; and
upsampling means for interpolating between selected pixels of different lines of said source data in the interpolation direction selected by said selection means to compute intermediate pixels on a line segment passing through
25 said upsampled target pixel and for interpolating between the intermediate pixels to generate said upsampled target pixel.

26. An apparatus as defined in claim 25 wherein said upsampling means computes pixels on a line segment parallel to the lines of said source data.

27. An apparatus as defined in claim 26 wherein said comparison means compares pixels of different lines of said source data in at least three different
5 directions.

28. An apparatus as defined in claim 27 wherein said comparison means compares the pixels of different lines of said source data in a vertical direction and in opposite oblique directions forming an angle with said vertical direction, said angle being in the range $0 < \text{angle} < 90^\circ$.

10 29. An apparatus as defined in claim 28 wherein said comparison means generates a difference value for each of said three comparisons and wherein said selection means compares the difference values with one another to determine whether the differences therebetween are within a threshold, said selection means selecting the interpolation direction associated with the smallest difference value
15 when none of the differences are within the threshold.

30. An apparatus as defined in claim 29 wherein said selection means selects the vertical interpolation direction when the differences between the all of the difference values are within said threshold.

20 31. An apparatus as defined in claim 30 wherein when the difference between the difference values in the vertical direction and only one of the oblique directions is within the threshold, the selection means selects that oblique interpolation.

25 32. An apparatus as defined in claim 31 wherein when the difference between the difference values in the two oblique directions is within the threshold, additional pixels of the different lines of source data are compared in the opposite oblique directions by the comparison means, the difference values generated as a

result of these comparisons being compared by said selection means, when the difference between the difference values generated by the selection means as a result of the comparison of the additional pixels is within the threshold, the vertical interpolation direction is selected by said selection means.

5 33. An apparatus as defined in claim 29 further comprising means to determine the closest pixel of said source data to said upsampled target pixel and the consecutive pixels in two lines of said source data that form the corners of a rectangular region surrounding said upsampled target pixel, one of the pixels of which is said closest pixel, said comparison means being responsive to said determination
10 means and comparing between the pixels that define said rectangular region.

34. An apparatus as defined in claim 27 wherein said comparison means compares pixels of different lines of said source data in at least seven different directions.

15 35. An apparatus as defined in claim 34 wherein said comparison means compares pixels of different lines of said source data in seven different directions, said seven different directions including a vertical direction and two sets of opposite oblique directions forming angles with said vertical direction, each set of opposite oblique directions including three oblique directions, each of said three oblique directions forming a different angle with said vertical direction, the compared pixels
20 in each set of three oblique directions being located on opposite sides of said upsampled target pixel.

36. An apparatus as defined in claim 35 wherein said comparison means compares pixels in oblique directions forming 26.6° , 45° and 63.4° angles with said vertical direction.

25 37. An apparatus as defined in claim 25 wherein said upsampling means includes a directional interpolator for computing said intermediate pixels and a second interpolator for computing said upsampled target pixel.

- 41 -

38. An apparatus as defined in claim 37 wherein each of said linear interpolators includes a pair of adders and a multiplier.

39. An apparatus as defined in claim 35 wherein said comparison means includes a plurality of subtractors, each subtractor being associated with one of said oblique directions.

40. An apparatus as defined in claim 37 wherein said upsampling means includes a pair of linear interpolators for computing said intermediate pixels and a third linear interpolation for computing said upsampled target pixel.

41. An apparatus as defined in claim 40 wherein said directional interpolator includes a plurality of polyphase filters, each of said polyphase filters receiving coefficients from a memory.

42. An apparatus as defined in claim 41 wherein said second interpolator includes a polyphase filter receiving coefficients from a second memory.

43. An apparatus as defined in claim 36 wherein said comparison means generates a difference value for each of said seven comparisons, the difference values for said oblique directions in each set being summed to yield a working difference value.

44. An apparatus as defined in claim 43 wherein said selection means compares the working difference values and the difference value associated with the vertical direction with one another to determine whether the differences therebetween are within a threshold, said selection means selecting a set of possible interpolation directions associated with the smallest difference value when none of the differences are within the threshold, the interpolation directions in said set being initially examined to eliminate interpolation directions which interpolate between pixels in a direction that crosses a boundary.

- 42 -

45. An apparatus as defined in claim 44 wherein said selection means selects the vertical interpolation direction when opposite oblique interpolation directions remain in the set of possible interpolation directions.

5 46. An apparatus as defined in claim 45 wherein if only one oblique interpolation direction remains in the set of possible interpolation directions, the selection means selects that oblique interpolation direction.

1/25

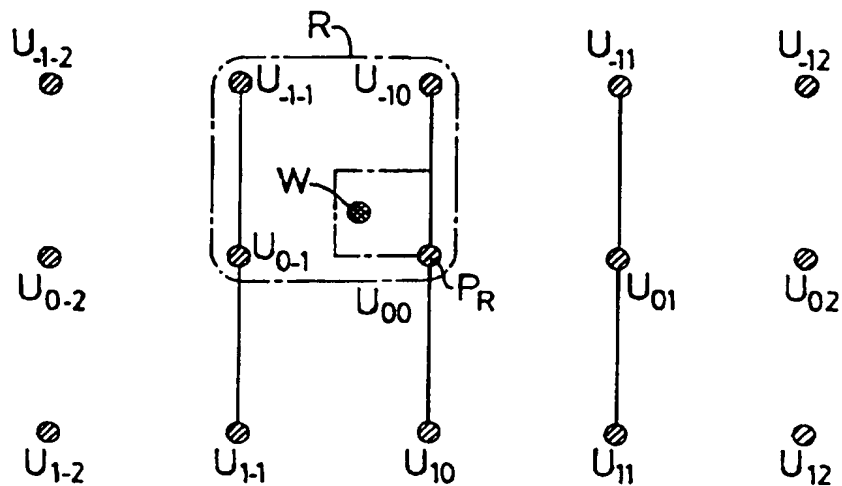


FIG. 1

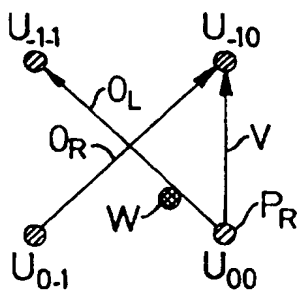


FIG. 2

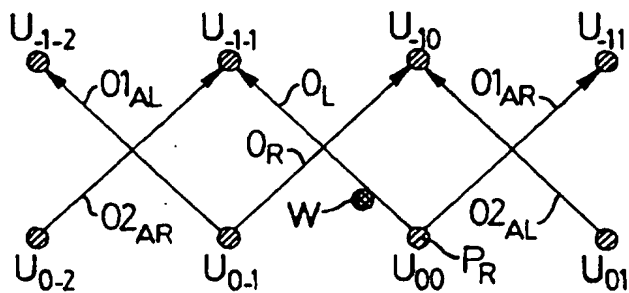


FIG. 3

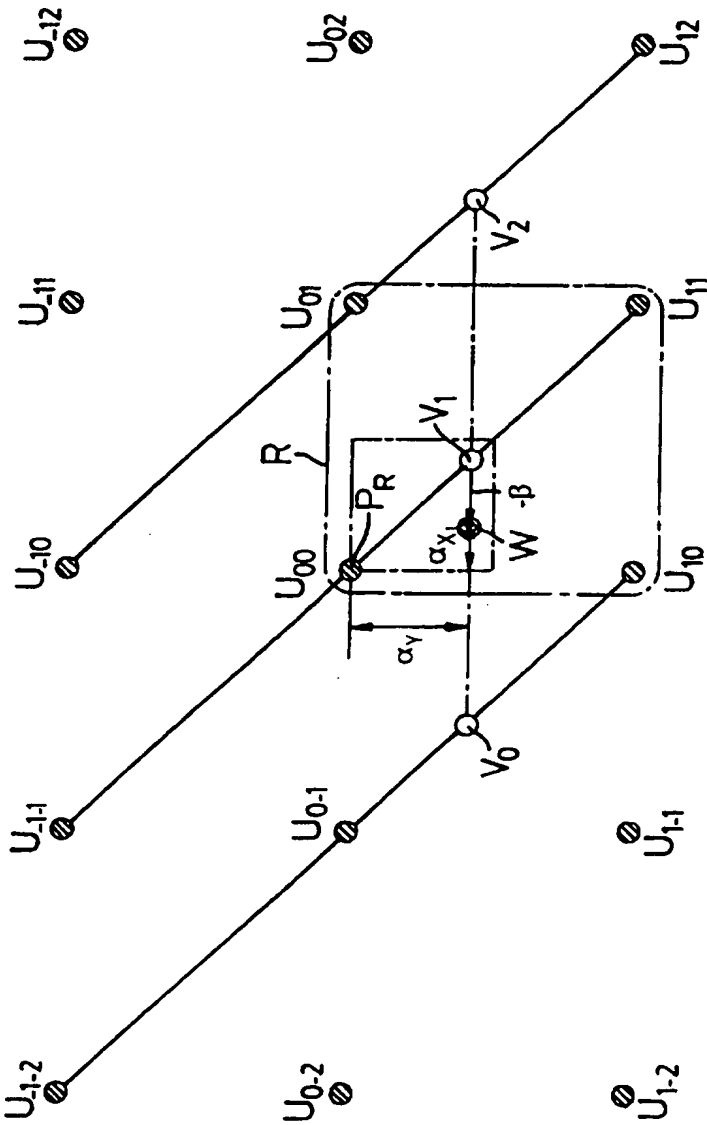


FIG. 5

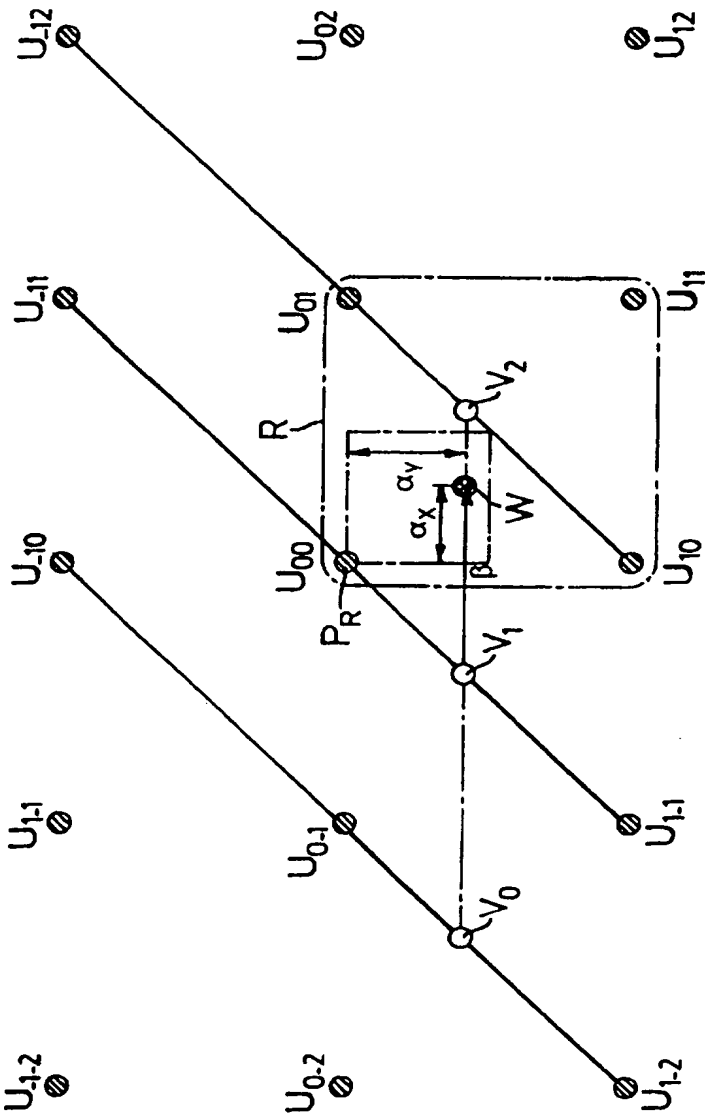
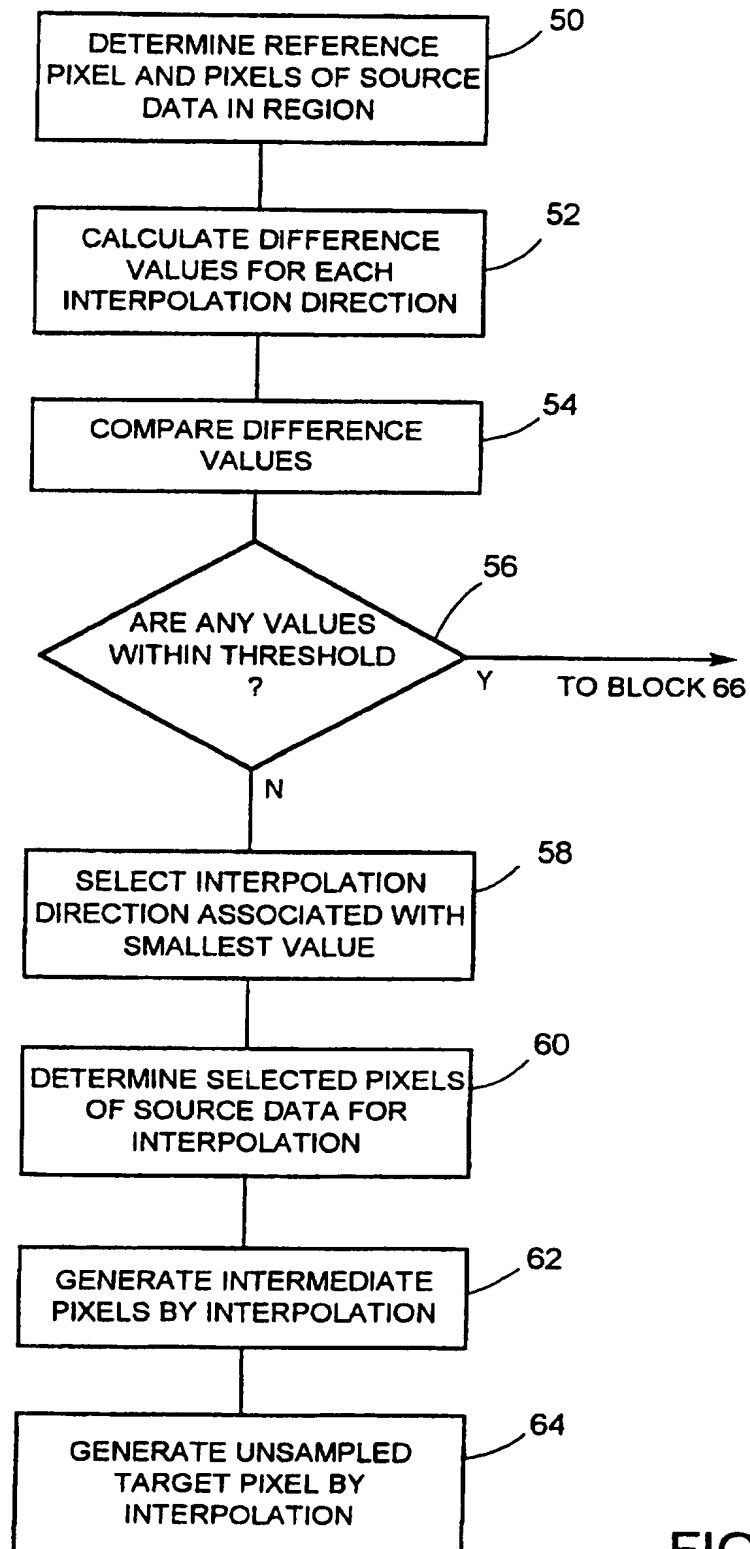


FIG. 6

5/25

FIG. 7A

6/25

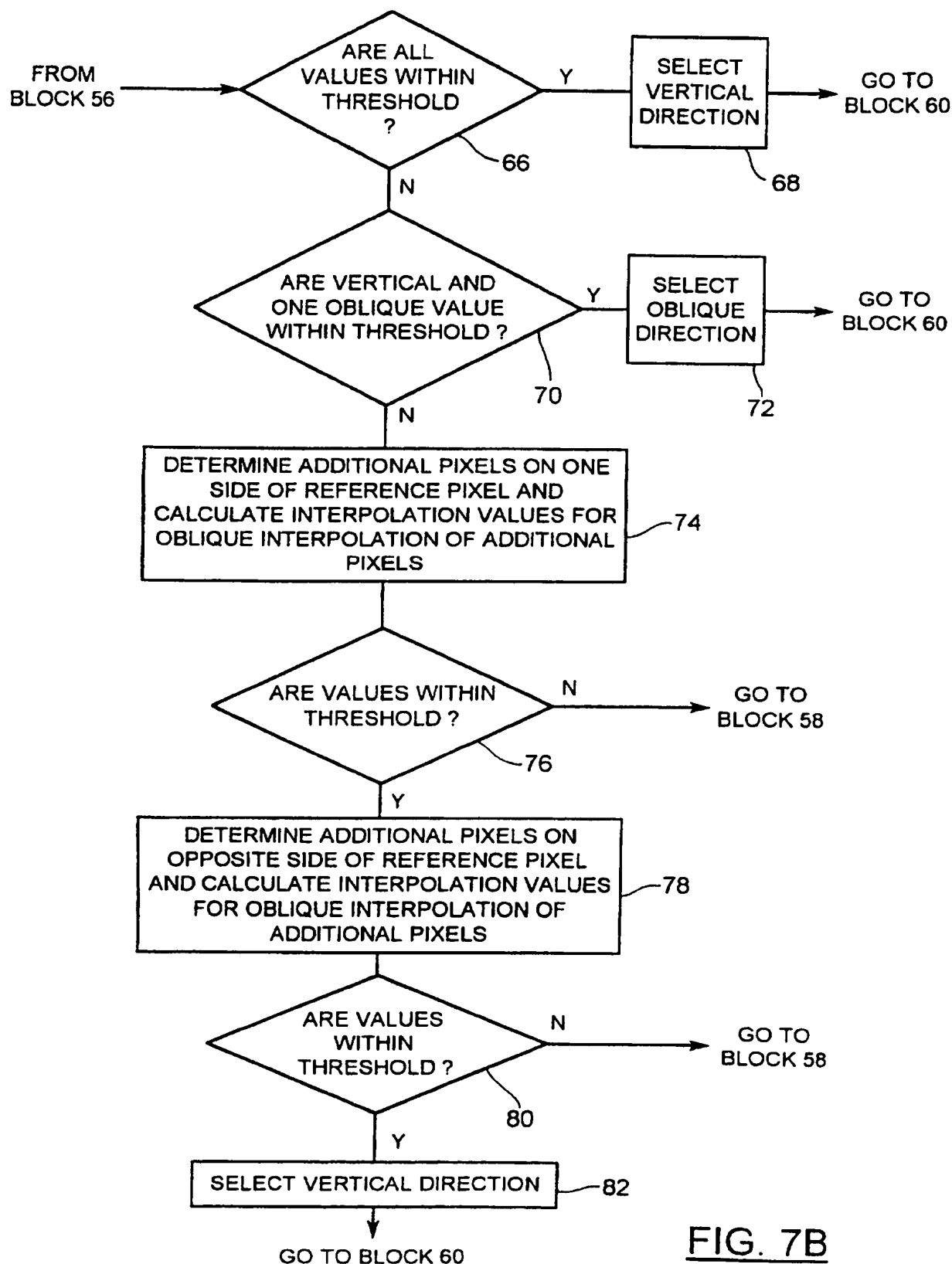
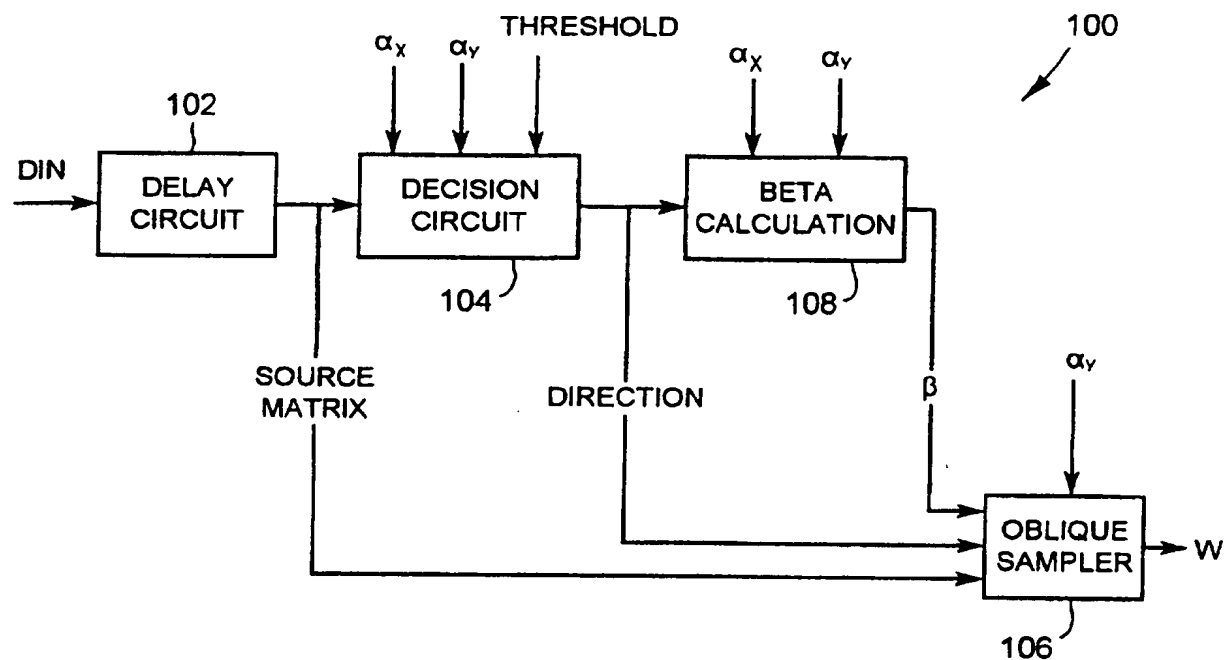
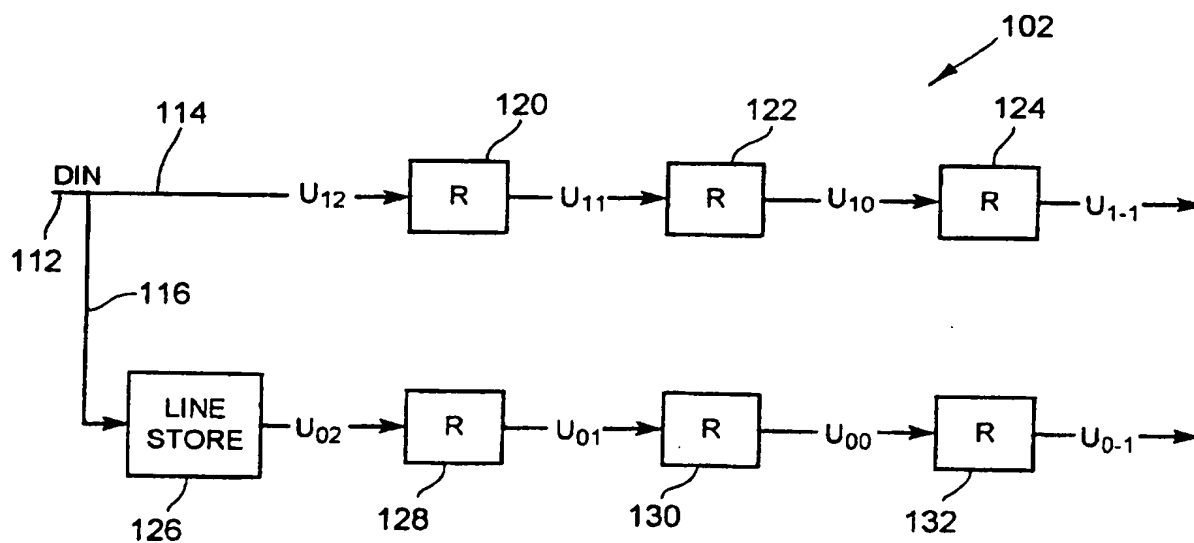
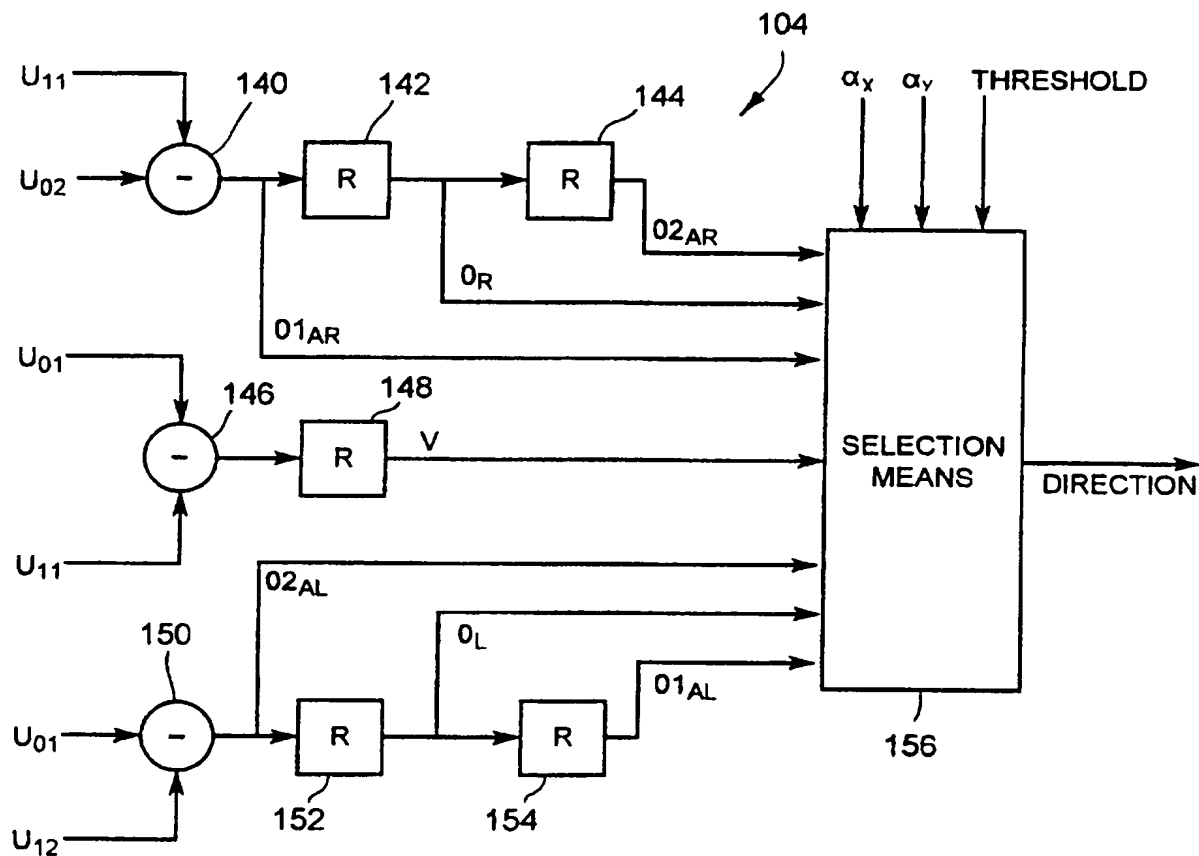
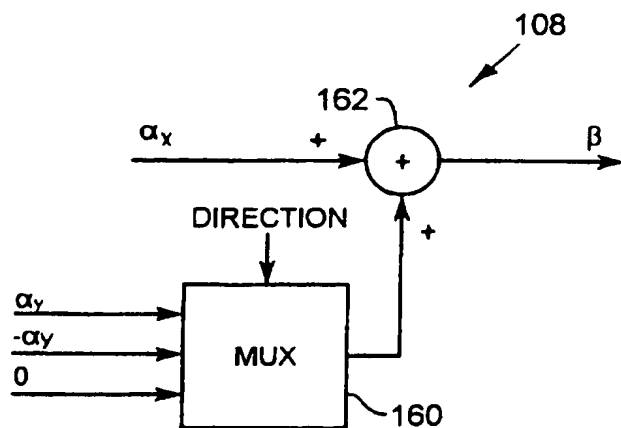


FIG. 7B

7/25

FIG. 8FIG. 9

8/25

FIG. 10FIG. 11

9/25

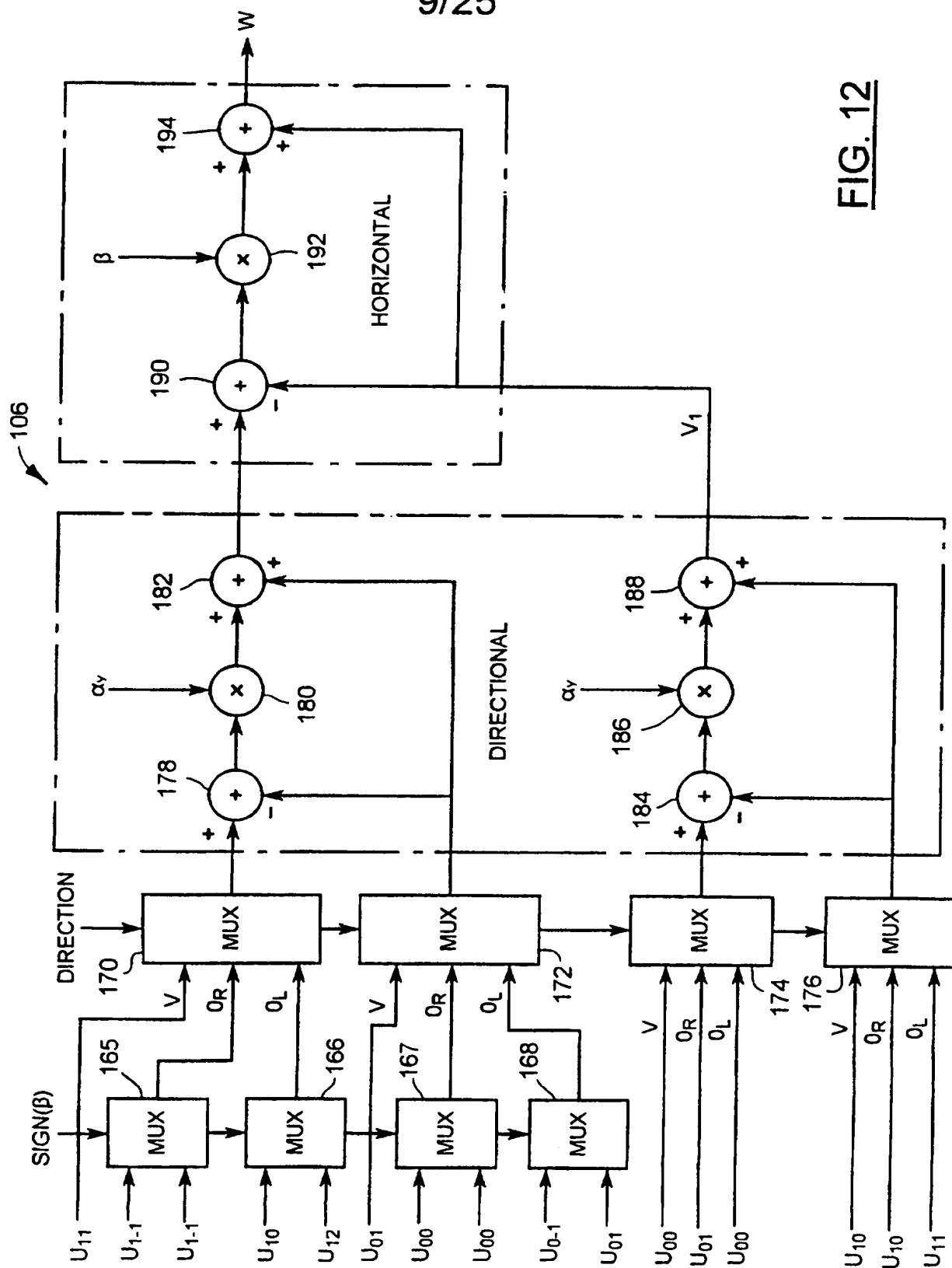


FIG. 12

10/25

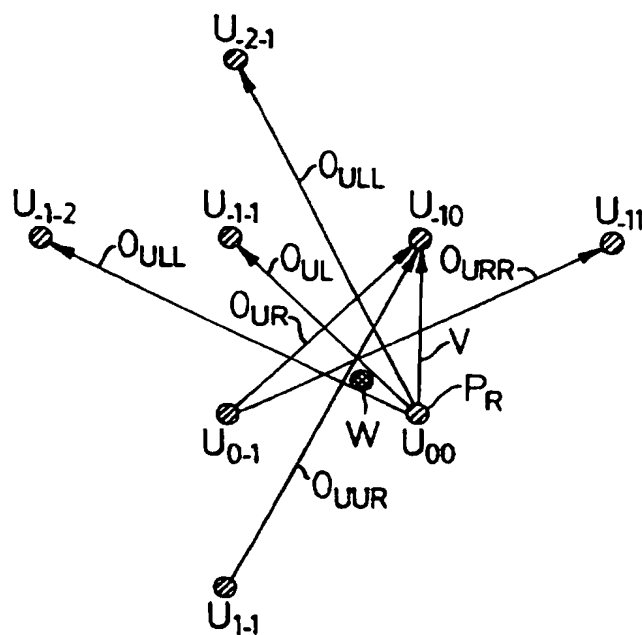


FIG. 13

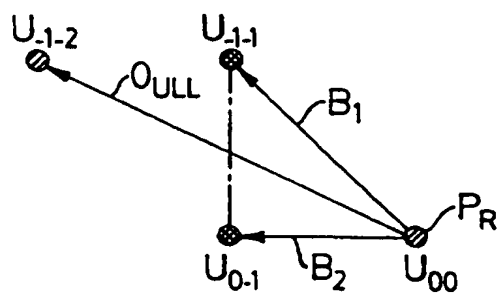


FIG. 14

11/25

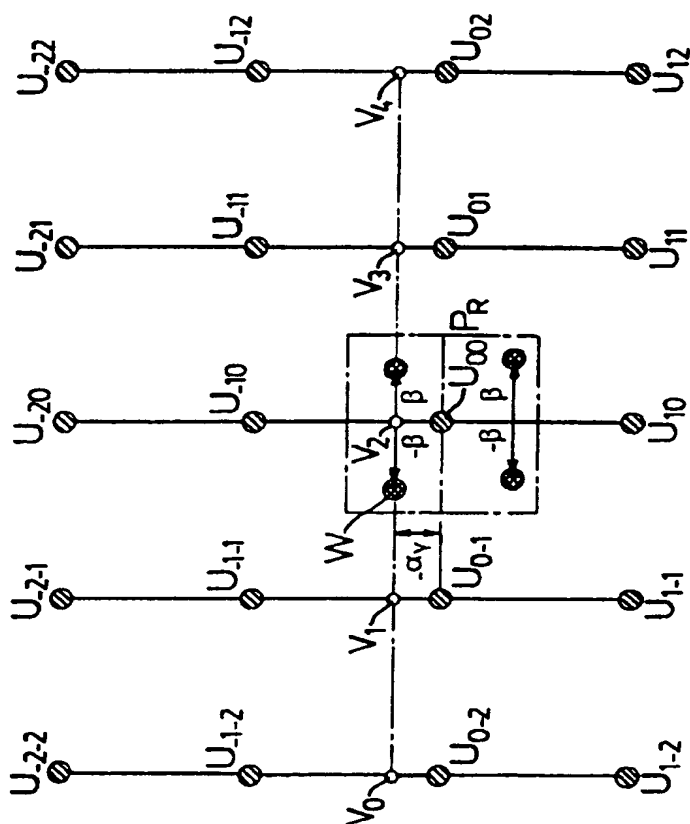


FIG. 15

12/25

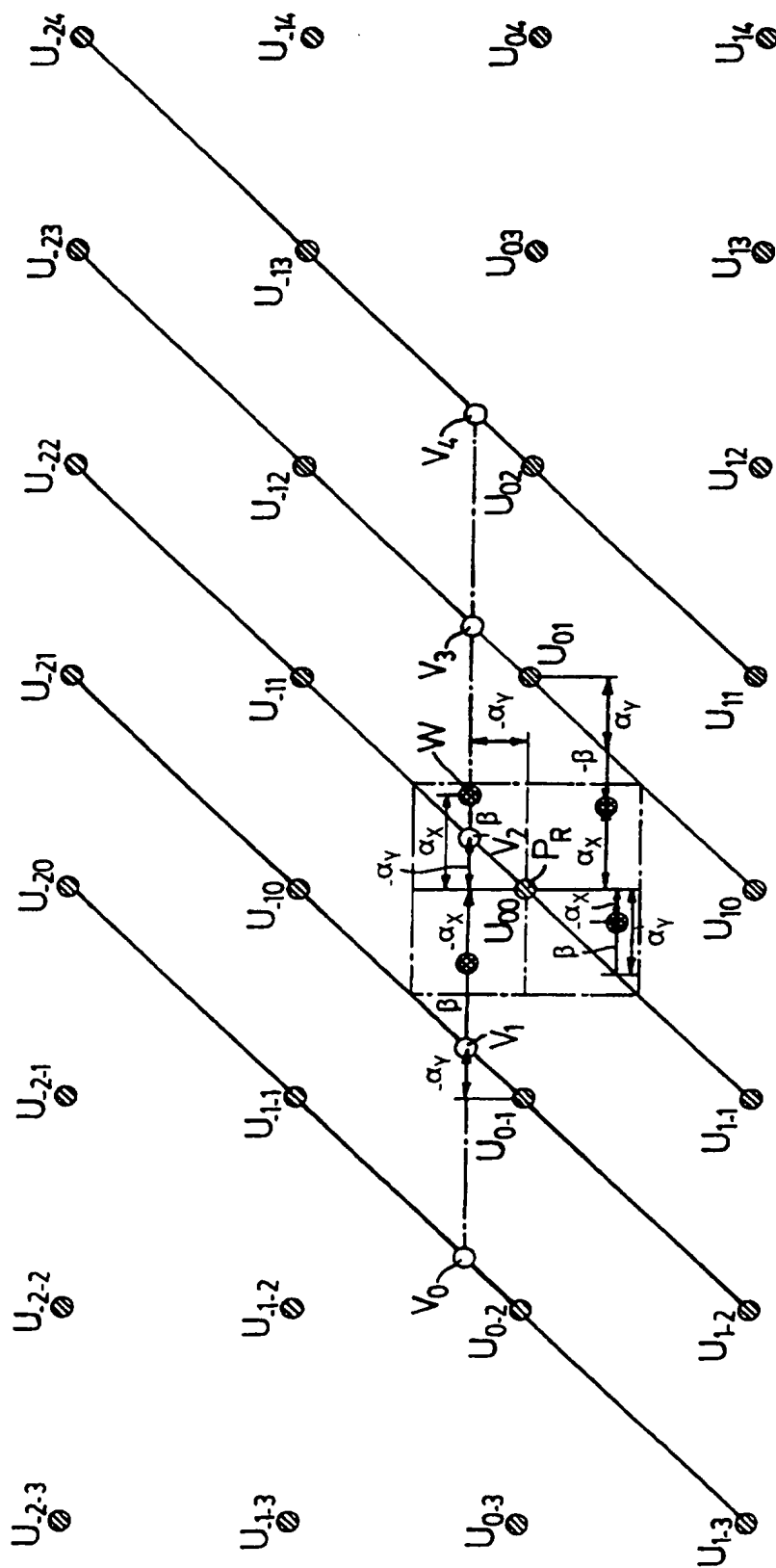


FIG. 16

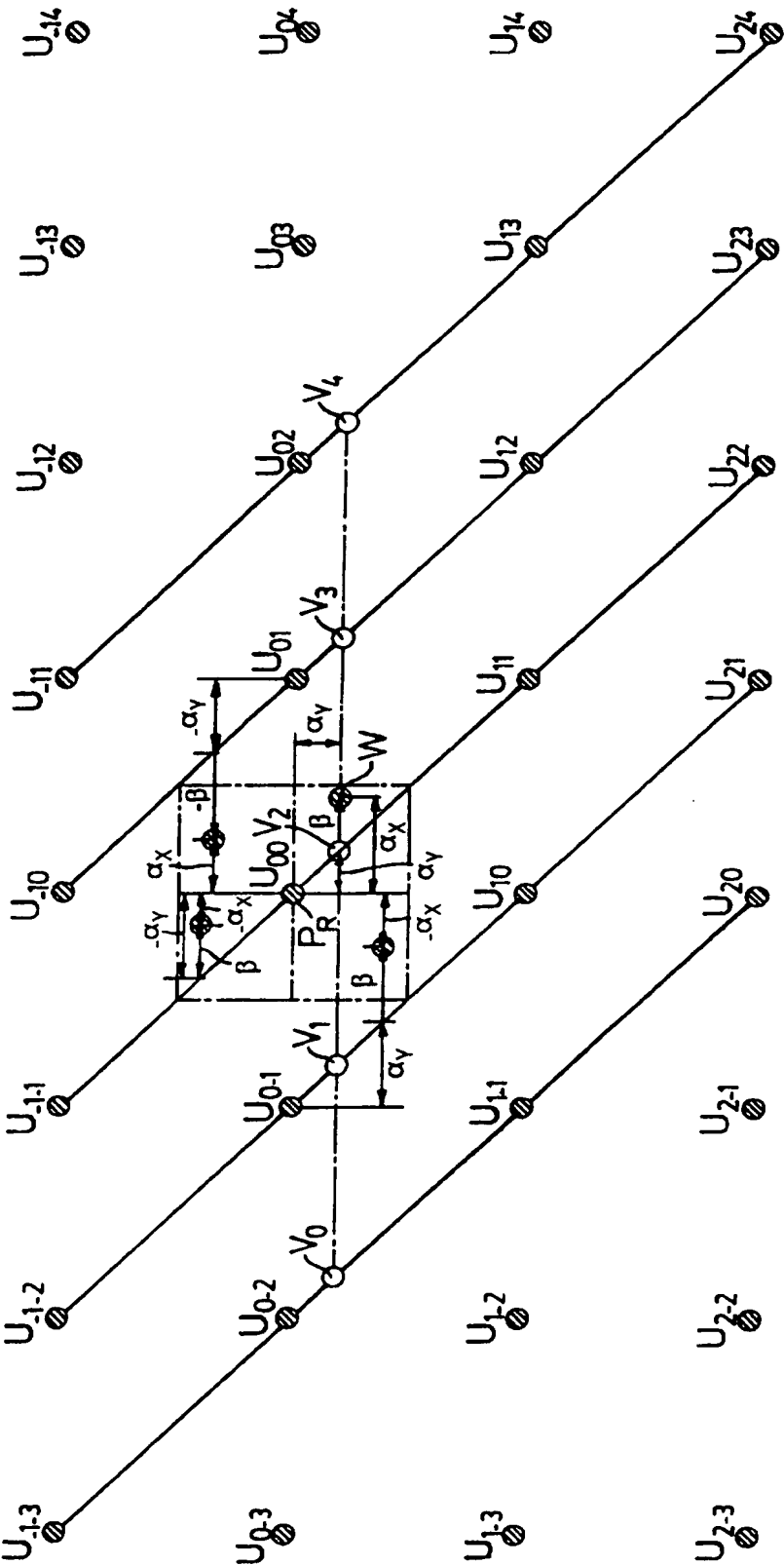


FIG. 17

14/25

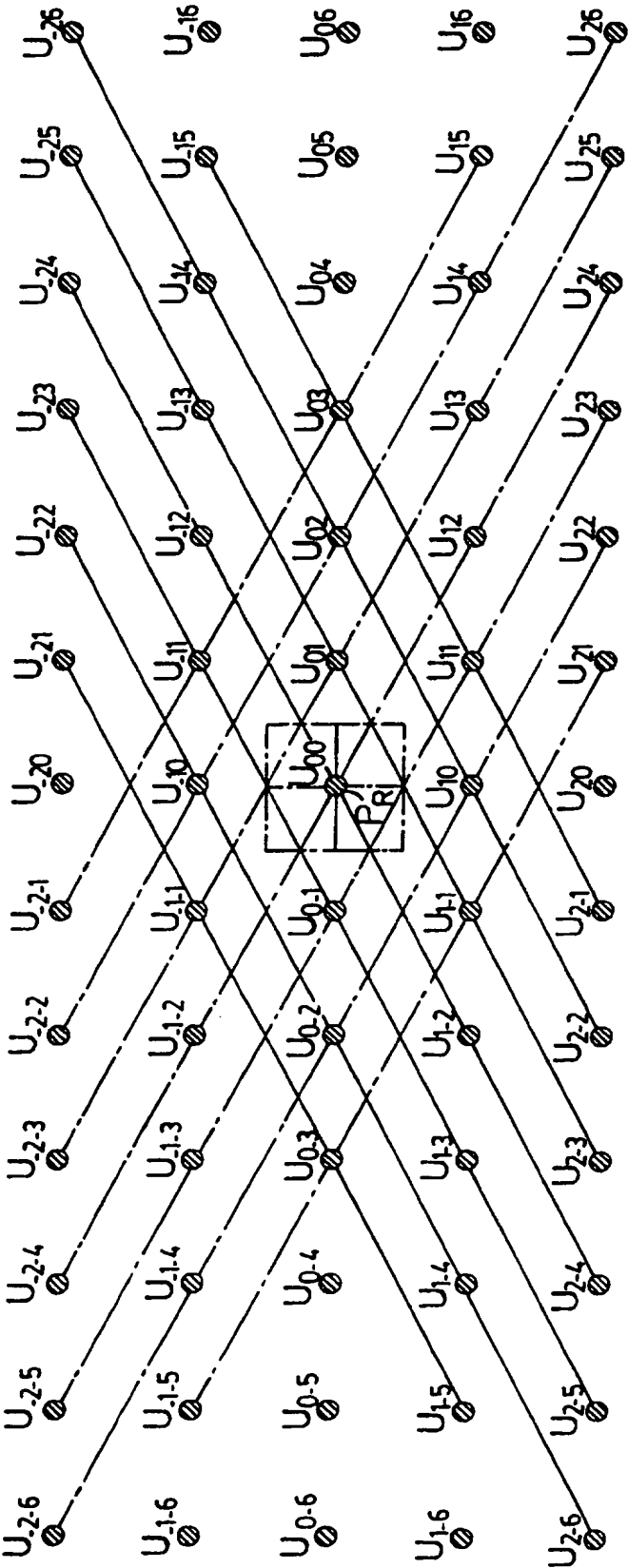


FIG. 18

15/25

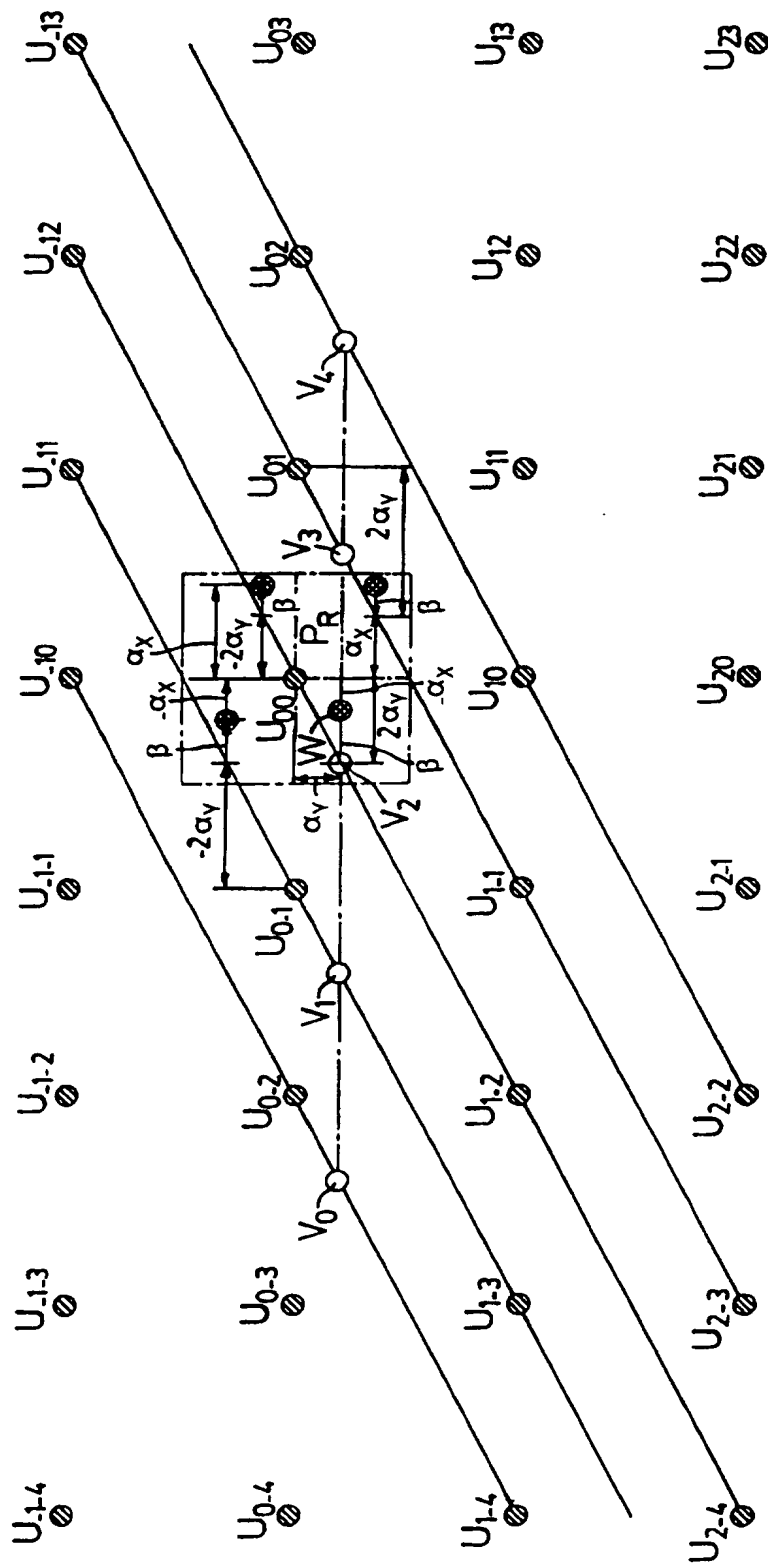


FIG. 19

16/25

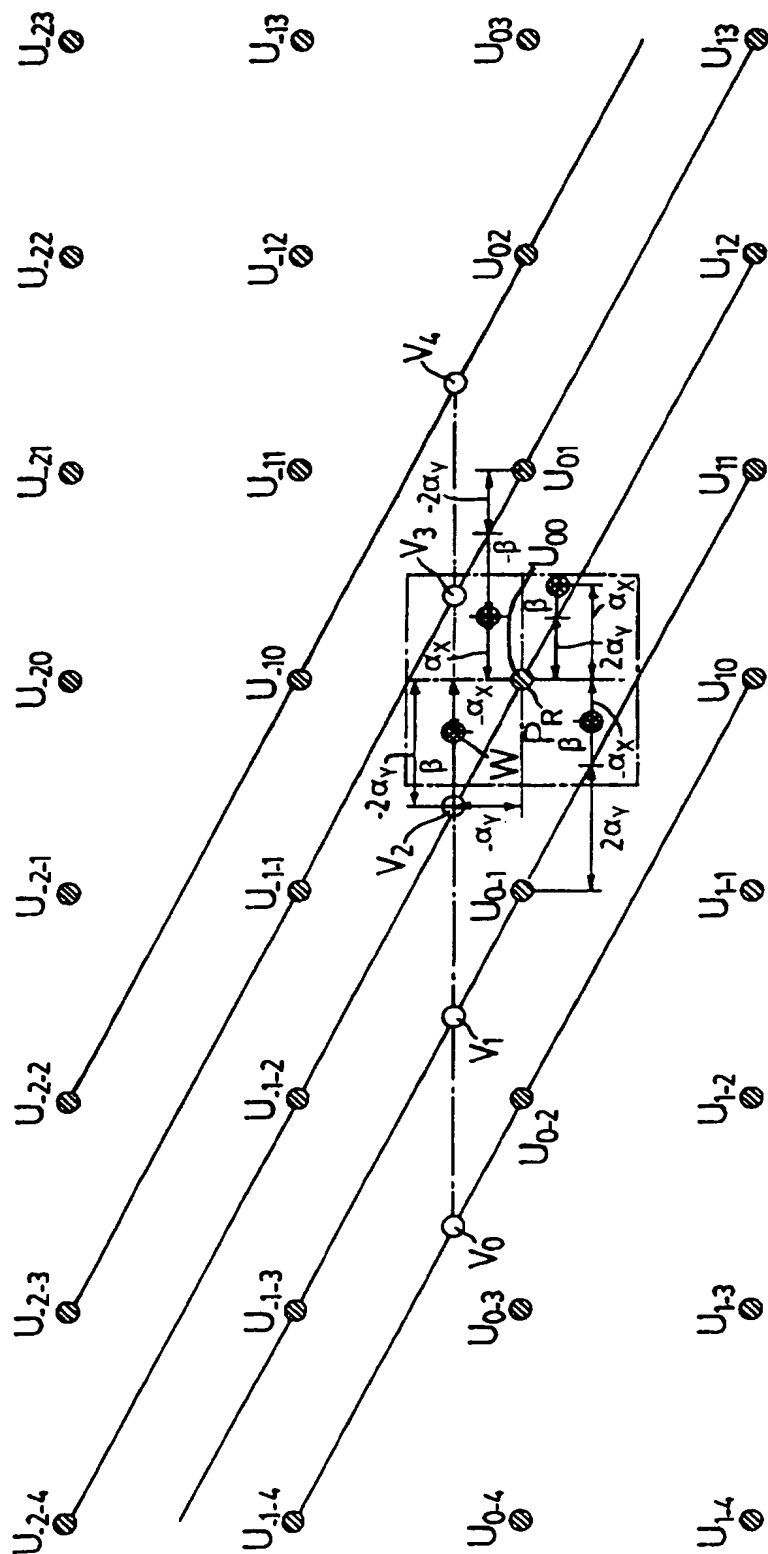


FIG. 20

17/25

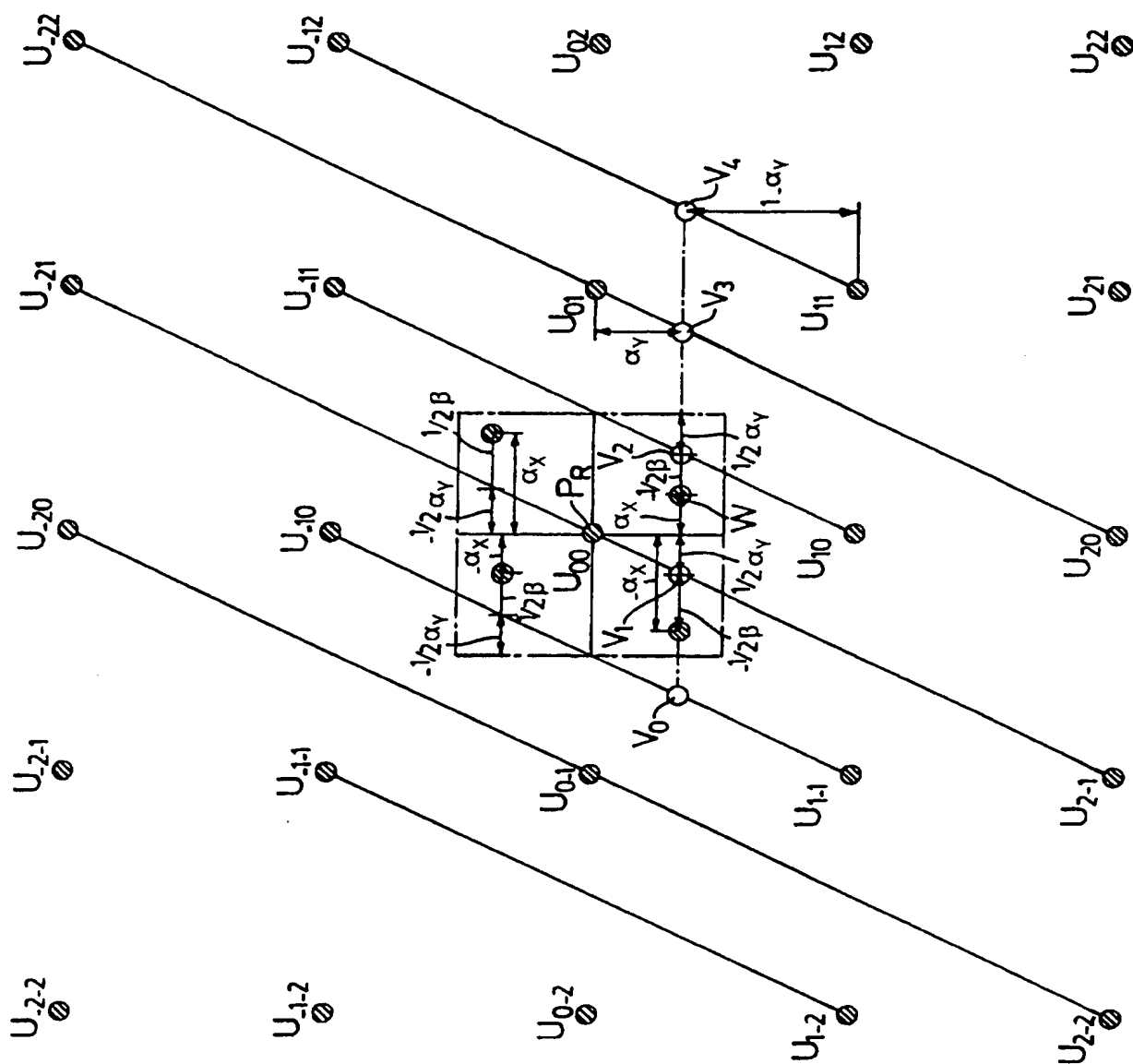


FIG. 21

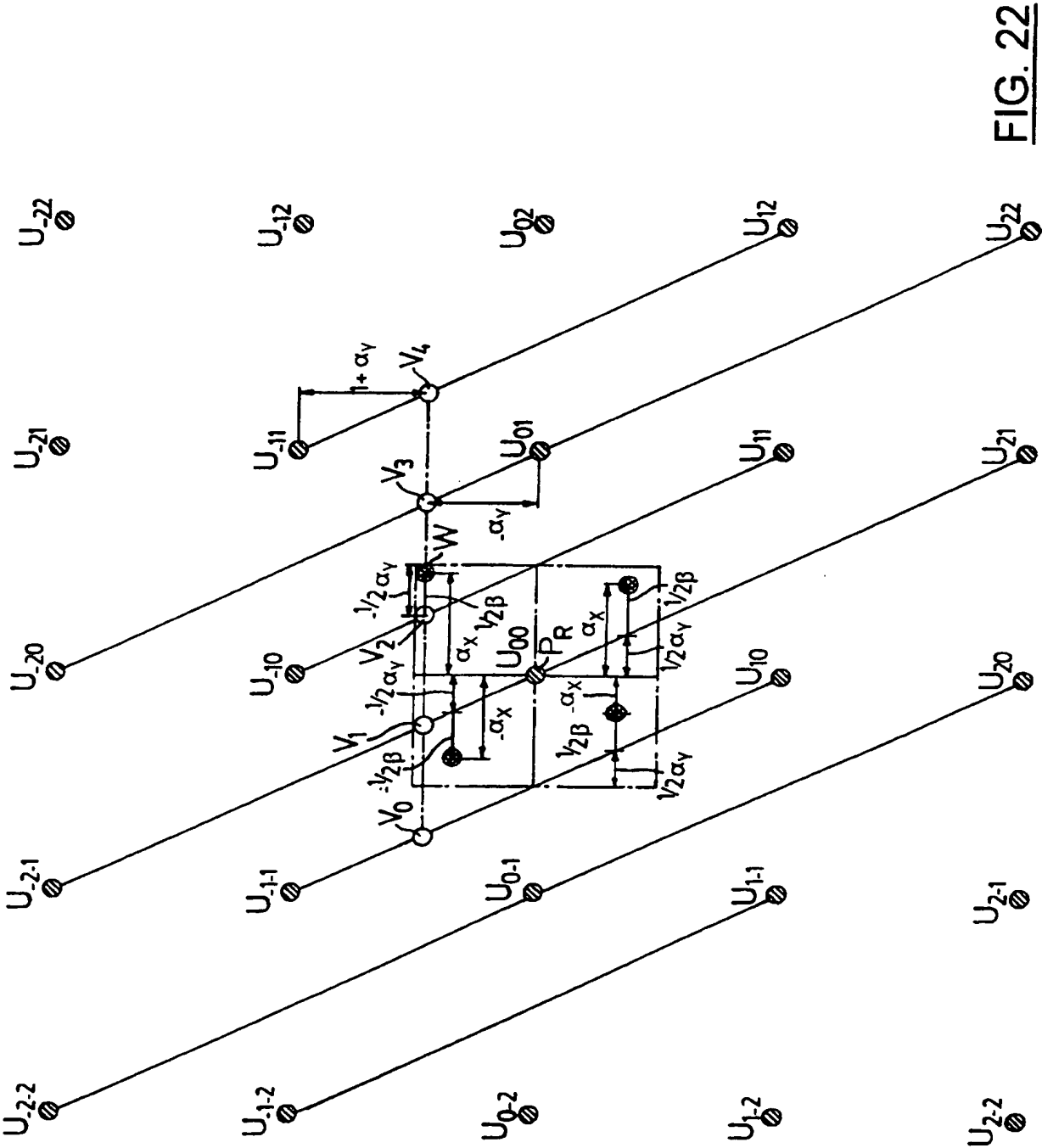


FIG. 22

19/25

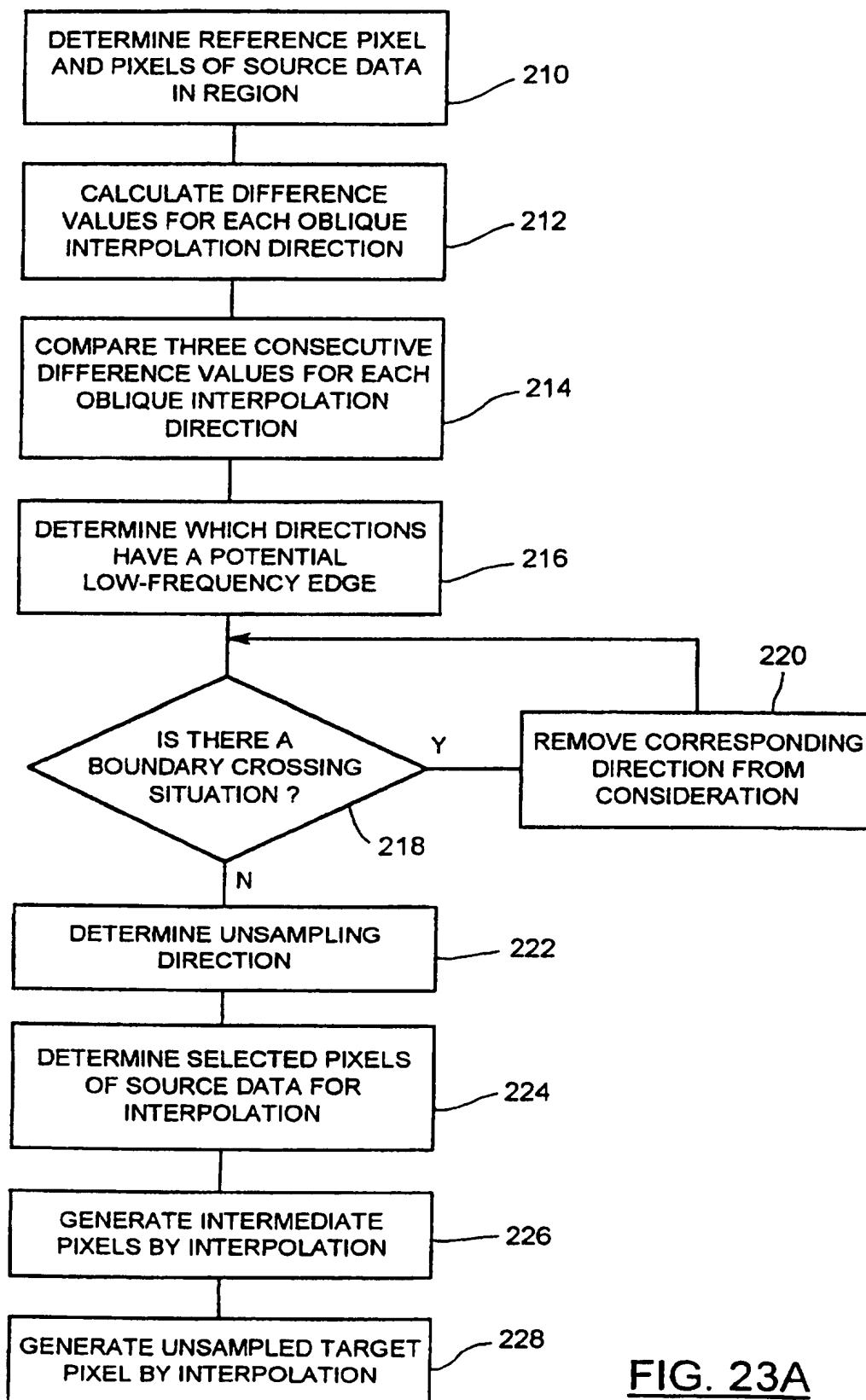


FIG. 23A

20/25

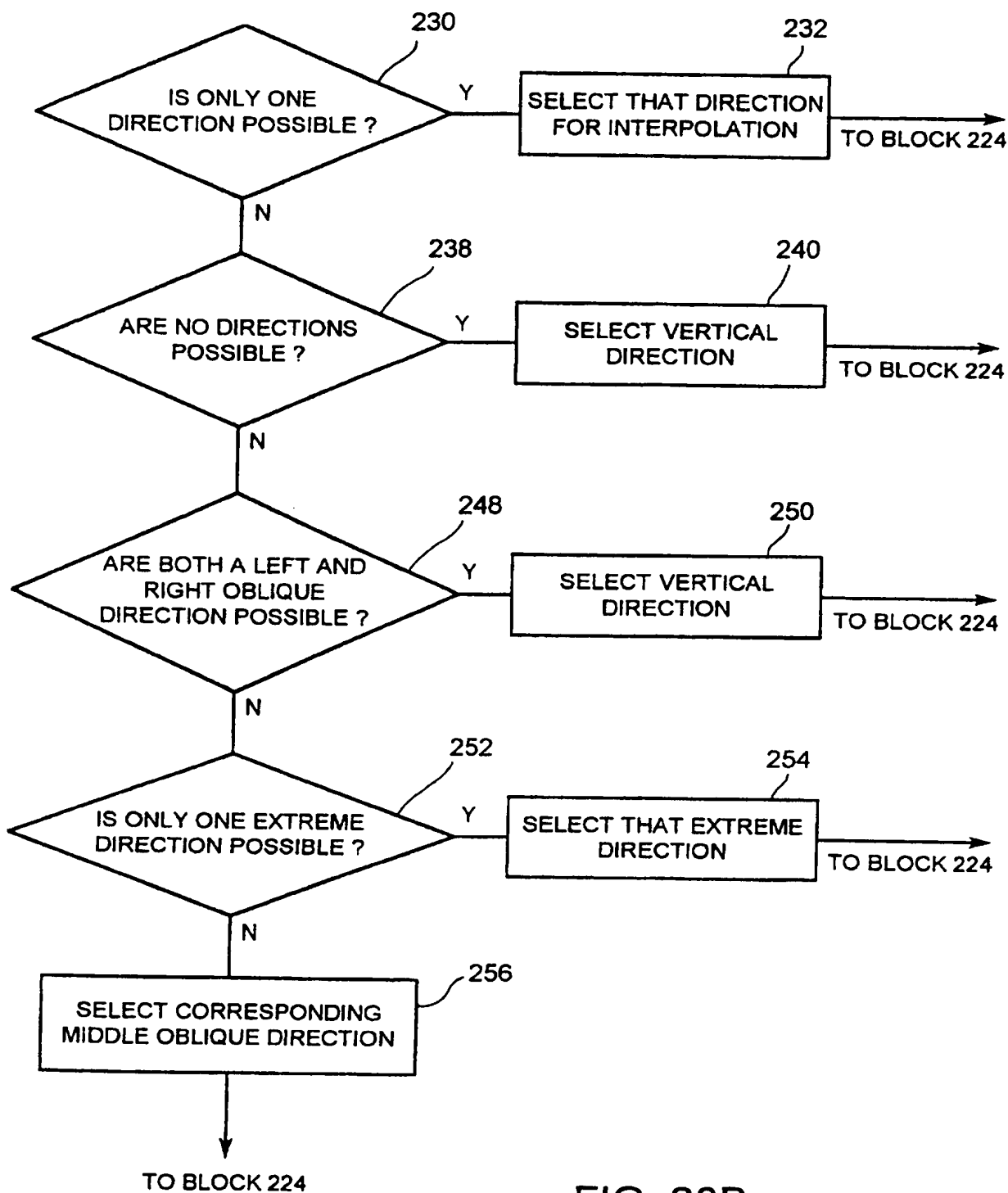
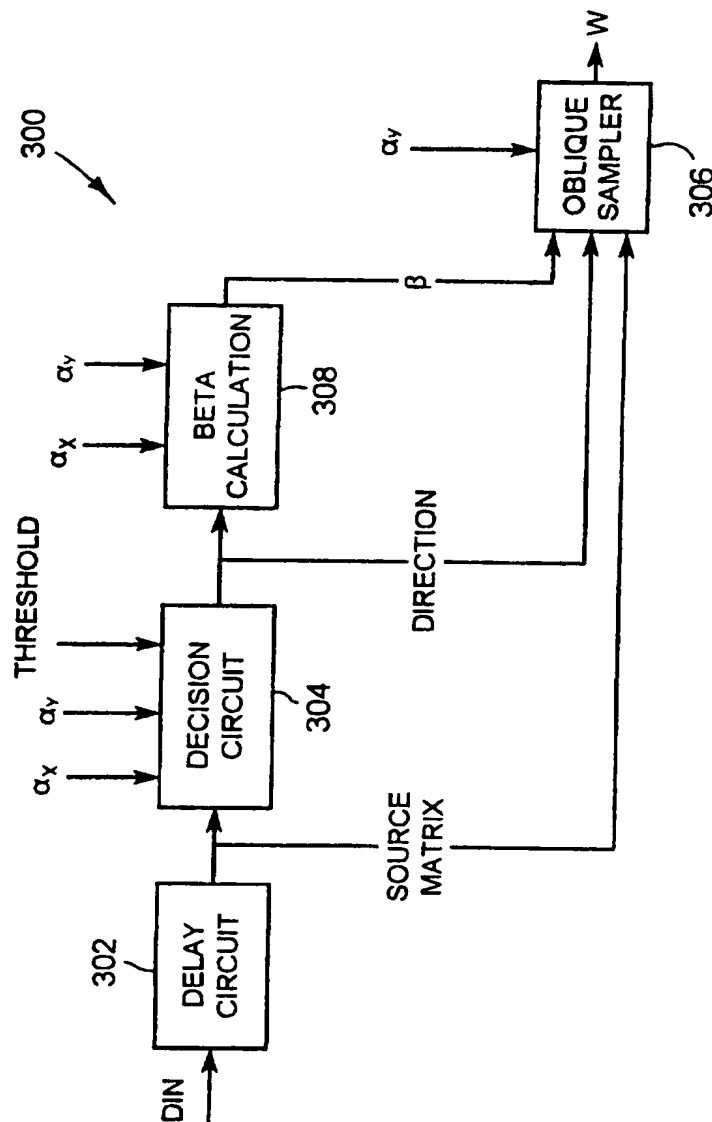


FIG. 23B

21/25

FIG. 24

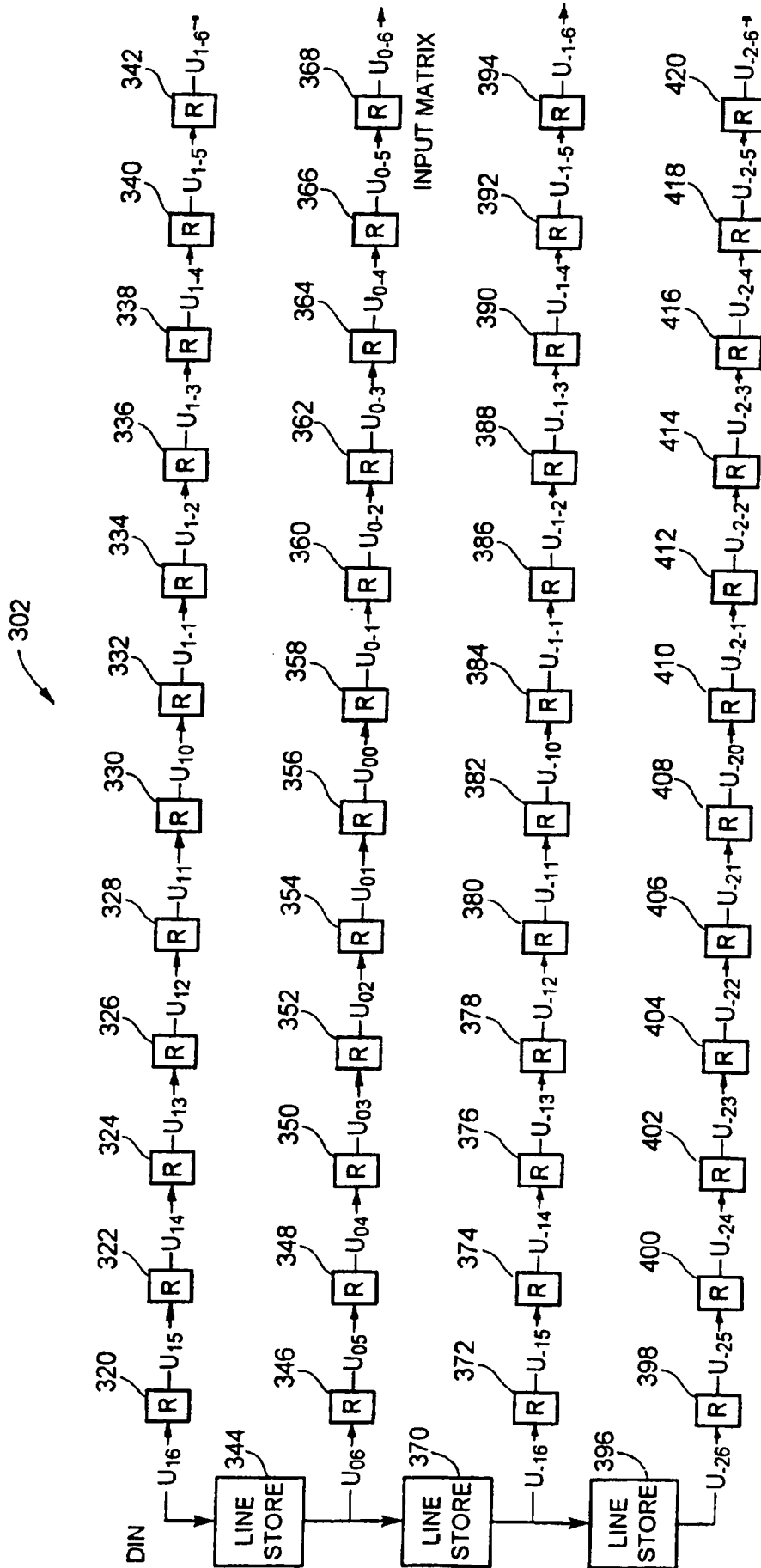


FIG. 25

23/25

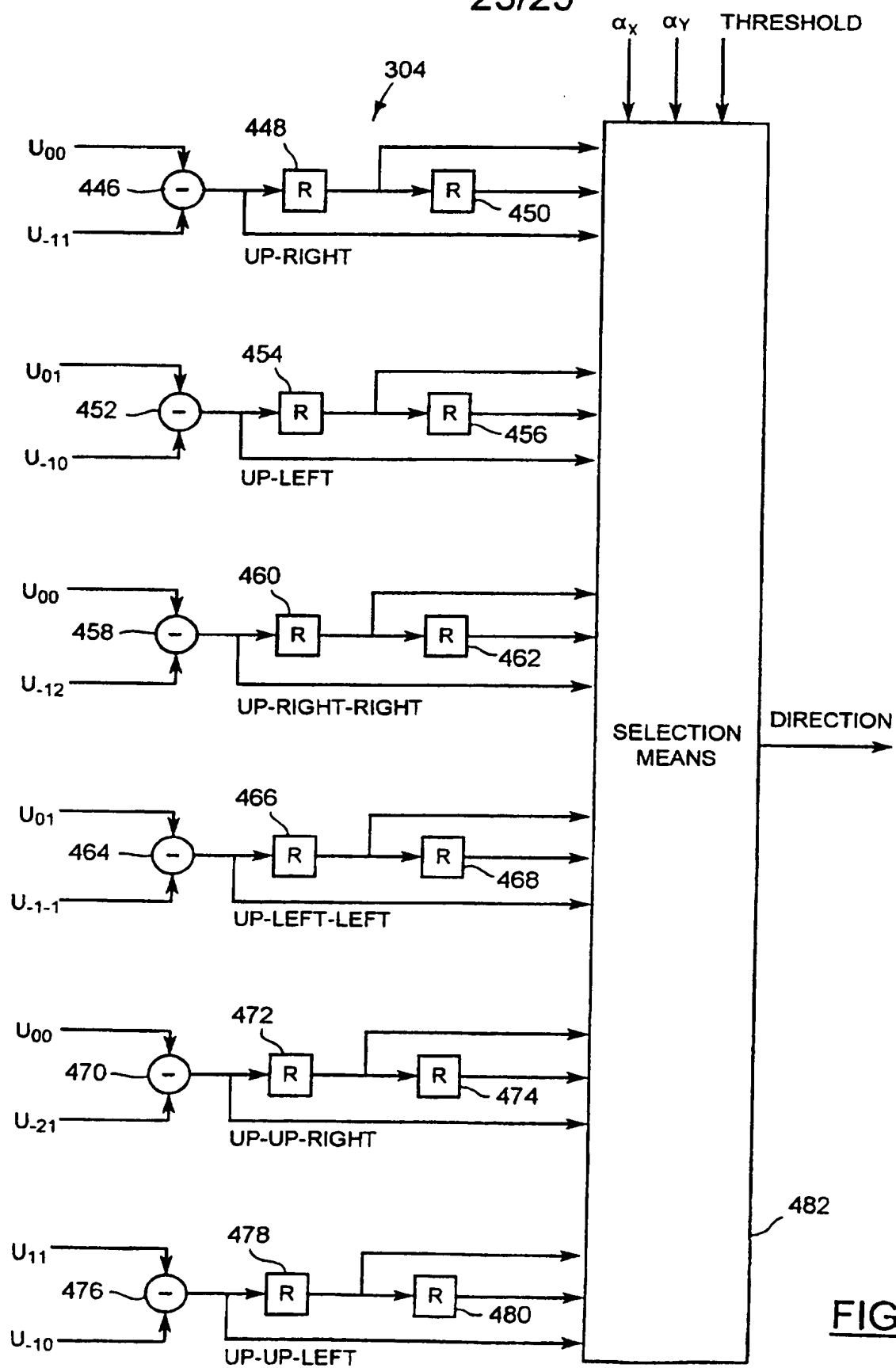
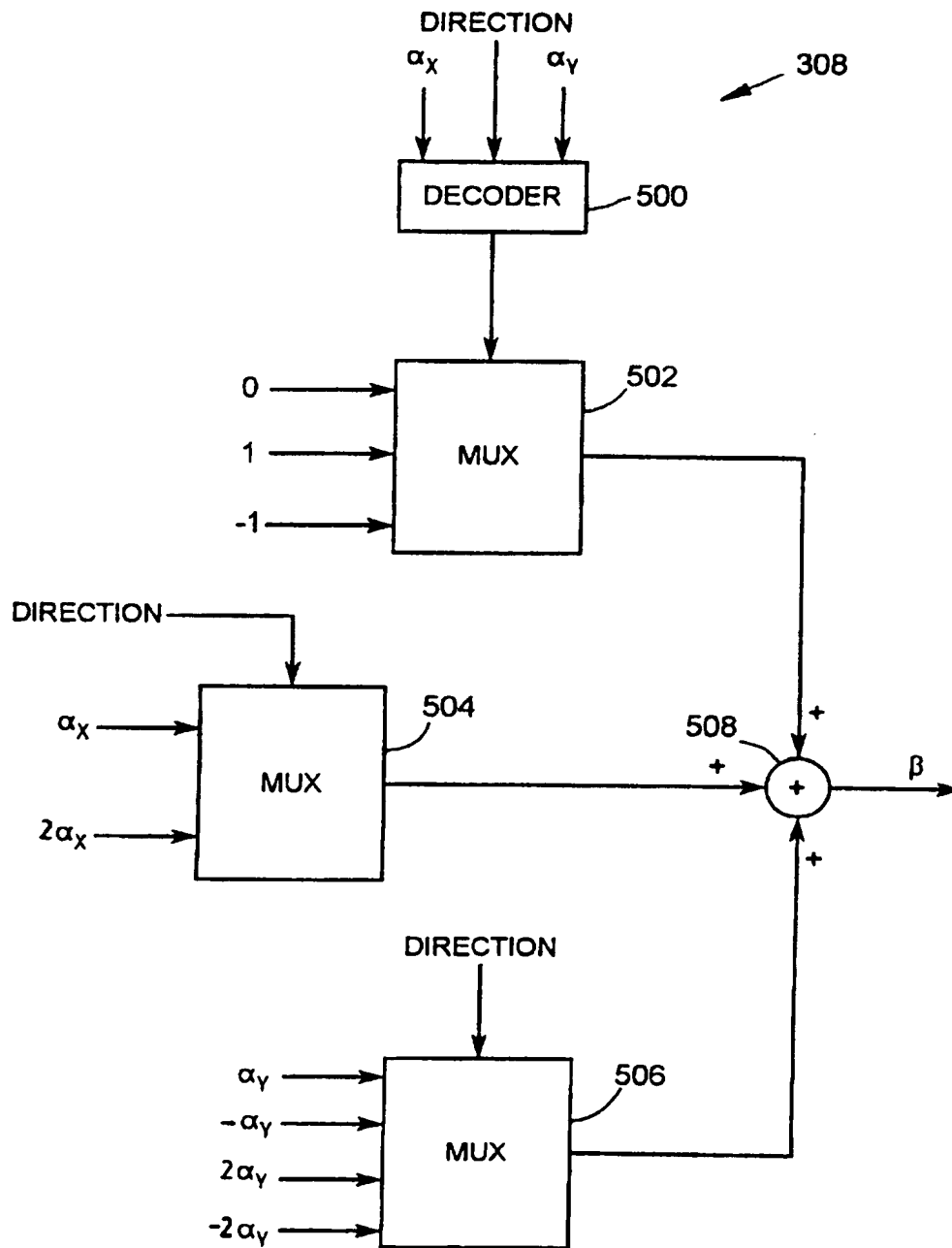


FIG. 26

SUBSTITUTE SHEET (RULE 26)

24/25

FIG. 27

25/25

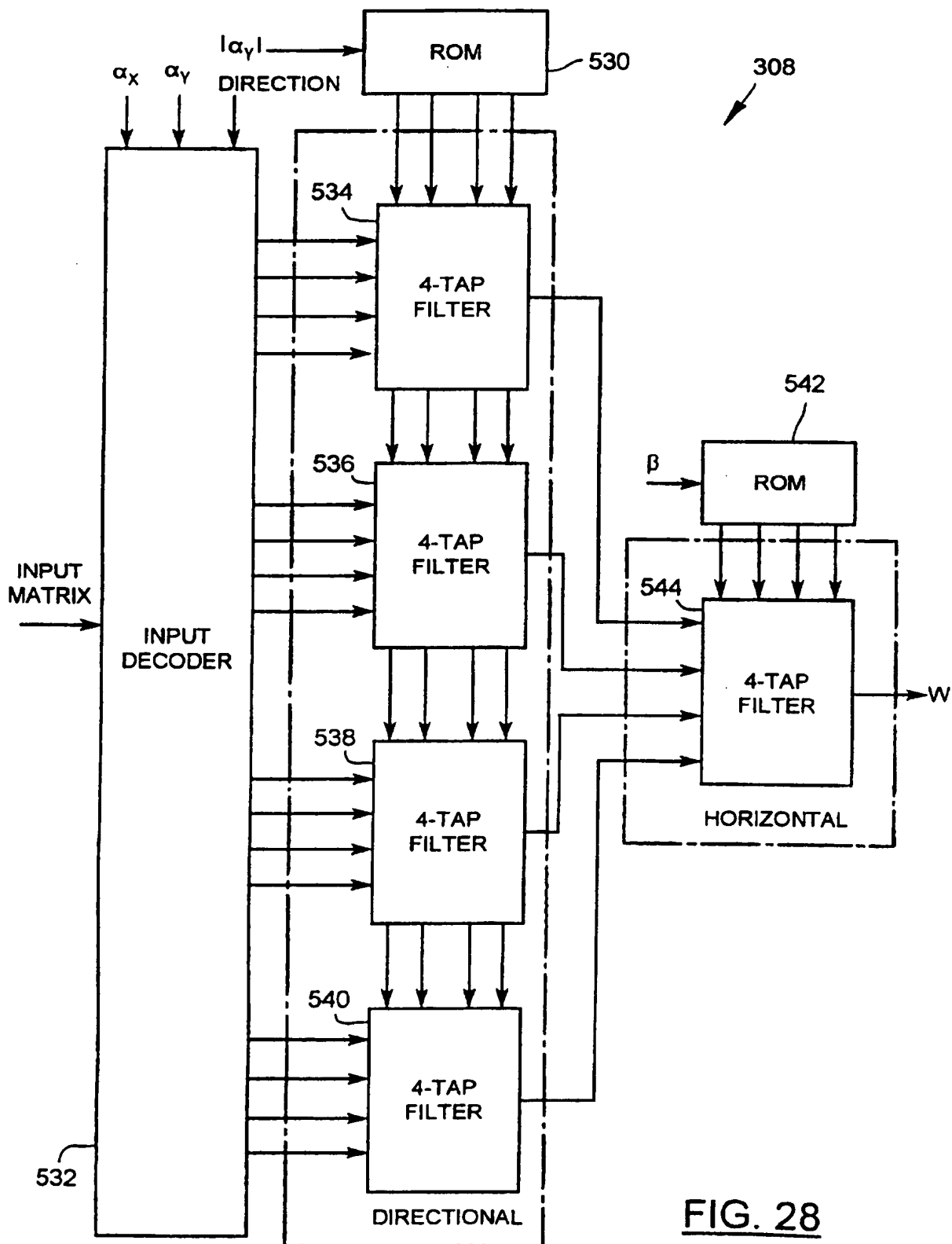


FIG. 28

INTERNATIONAL SEARCH REPORT

Intern al Application No
PCT/CA 96/00703

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04N5/262

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04N G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	DE,A,44 07 101 (SIEMENS AG) 14 September 1995 see column 2, line 34 - line 46 see column 3, line 53 - line 62 see column 4, line 6 - column 5, line 59 ---	1,25
A	EP,A,0 550 231 (SALON TELEVISIOTEHDAS OY) 7 July 1993 see page 2, line 26 - page 3, line 54 see page 4, line 32 - page 7, line 7 ---	1-46
A	EP,A,0 561 286 (THOMSON CONSUMER ELECTRONICS) 22 September 1993 see column 1, line 11 - line 53 see column 3, line 45 - column 4, line 20 --- -/--	1,25

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "&" document member of the same patent family

Date of the actual completion of the international search

20 January 1997

Date of mailing of the international search report

- 6. 02. 97

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl,
Fax (+ 31-70) 340-3016

Authorized officer

Wentzel, J

INTERNATIONAL SEARCH REPORT

International Application No

PCT/CA 96/00703

C(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, vol. 39, no. 3, 1 August 1993, pages 234-240, XP000396285 SIMONETTI R ET AL: "A DEINTERLAVER FOR IQTV RECEIVERS AND MULTIMEDIA APPLICATIONS" see page 234, left-hand column, line 26 - page 236, right-hand column, last line see figure 5</p> <p>---</p>	1-24
A	<p>US,A,4 985 764 (SATO KOICHI) 15 January 1991 see column 2, line 17 - line 68 see column 3, line 58 - column 10, line 62</p> <p>---</p>	25
A	<p>SIGNAL PROCESSING OF HDTV, 2, TURIN, AUG. 30 - SEPT. 1, 1989, no. WORKSHOP 3, 30 August 1989, CHIARIGLIONE L, pages 711-721, XP000215289 DOYLE T ET AL: "PROGRESSIVE SCAN CONVERSION USING EDGE INFORMATION"</p> <p>-----</p>	

INTERNATIONAL SEARCH REPORT

information on patent family members

International Application No

PCT/CA 96/00703

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
DE-A-4407101	14-09-95	FR-A- 2717027 GB-A- 2287154 US-A- 5579053	08-09-95 06-09-95 26-11-96
EP-A-0550231	07-07-93	NONE	
EP-A-0561286	22-09-93	CA-A- 2090696 US-A- 5481311	21-09-93 02-01-96
US-A-4985764	15-01-91	JP-A- 2177683	10-07-90